Automation of the Dipole Subtraction Method

Dissertation

zur

Erlangung der naturwissenschaftlichen Doktorwürde
(Dr. sc. nat.)

vorgelegt

der Mathematisch-naturwissenschaftlichen Fakultät

der

Universität Zürich

von

Nicolas Greiner

aus

Deutschland

Promotionskomitee

Prof. Dr. Thomas Gehrmann (Vorsitz und Leitung)
Prof. Dr. Ben Moore
Prof. Dr. Daniel Wyler

Zürich, 2010

Abstract

Numerical calculations of higher order corrections in perturbation theory require a systematic treatment of infrared divergencies which are an inherent property of the underlying quantum theory.
In this thesis we present a way of automating the construction and calculation of the subtraction terms following the approach of the dipole formalism. Besides the discussion of the fundamental aspects of this formalism we describe how this method has been implemented in the MadGraph/MadEvent package and what has lead to the development of the MadDipole package. We have applied the automation both on the massless and the massive case and therefore obtained a very general tool for an arbitrary process for next-to-leading order calculations in QCD.
As an application we have calculated the next-to-leading order corrections to the production of four b-quarks at the LHC. The Standard Model contribution to this process is the background to possible signals coming from supersymmetric theories. In this thesis we restrict ourselves to the case with only quarks in the initial state. We demonstrate that the inclusion of NLO corrections yields to a stabilization of the perturbative prediction. We consider a vriety of kinematical distributions among which we identify those whose shapes are most stable under QCD corrections.

Zusammenfassung

Numerische Berechnungen von Korrekturen in höherer Ordnung Störungstheorie erfordern eine systematische Behandlung von infraroten Singularitäten, die eine innewohnende Eigenschaft der zugrunde liegenden Quantentheorie sind.
In dieser Doktorarbeit präsentieren wir einen Weg die Konstruktion und Berechnung der Subtraktionsterme zu automatisieren, wobei wir dem Zugang des Dipolformalismus folgen. Neben der Diskussion der fundamentalen Aspekte dieses Formalismus beschreiben wir, wie diese Methode in das MadGraph/MadEvent Paket eingebaut wurde, was zu der Entwicklung des MadDipole Pakets geführt hat. Dabei haben wir die Automatisierung sowohl für den masselosen als auch den massiven Fall durchgeführt wodurch man ein sehr allgemeine Werkzeug erhält, das für einen beliebigen Prozess für die nächst führende Ordnung in QCD angewendet werden kann.
Als eine Anwendung haben wir die nächst führenden Korrekturen zu der Produktion von vier b-Quarks am LHC berechnet. Der Beitrag des Standardmodells zu diesem Prozess ist der Hintergrund zu möglichen Signalen, die von supersymmetrischen Theorien kommen können. In dieser Doktorarbeit beschränken wir uns auf den Fall mit ausschließlich Quarks im Anfangszustand. Wir zeigen, dass die Hinzunahme von NLO Korrekturen zu einer Stabilisierung der störungstheoretischen Vorhersage führt. Wir betrachten eine Vielzahl von kinematischen Verteilungen. Von diesen Verteilungen indentifizieren wir diejenigen, deren Form am stabilsten unter QCD Korrekturen sind.

Contents

Chapter 1

Introduction

... although the symmetries are hidden from us, we can sense that they are latent in nature, governing everything about us. That's the most exciting idea I know: that nature is much simpler than it looks. Nothing makes me more hopeful that our generation of human beings may actually hold the key to the universe in our hands - that perhaps in our lifetimes we may be able to tell why all of what we see in this immense universe of galaxies and particles is logically inevitable.

–S. Weinberg

Particle physics is the part of natural science that searches for the fundamental forces and interactions occurring in nature. The aim of particle physicists is to understand not only which particles exist in nature and can be produced in an experiment but also how they interact with each other. This requires an interplay between experimental physics on the one hand and theoretical physics on the other hand because one always needs both parts to achieve a success.

The recent startup of the large hadron collider (LHC) opens the door to new insight and gives hope for the discovery of new physics beyond the so called Standard Model of particle physics. With the LHC particle physics has entered a new era with a great discovery potential that may revolutionize our ideas of the microscopic world and its fundamental interactions.

During the last decades increasing efforts have been required to extend our limits of knowledge. But this has also had great impact on other fields leading to new developments both in experimental and theoretical physics. To mention only a few, this includes better electronics and hardware and new mathematical methods and efficient algorithms dealing with a huge amount of data.

Despite the great successes that have been achieved in particle physics there still remain many unanswered questions. The origin of mass and the resulting Higgs boson is probably the most famous one. In addition the question of whether we will find physics beyond the Standard Model and of what origin that will be is a very important and interesting query. And since the results of the WMAP satellite questions of particle physics are closely related to urgent questions in astrophysics

concerning the origin and nature of dark matter and dark energy.

1.1 Standard Model of particle physics

According to today's knowledge there exist four fundamental forces in nature: The electromagnetic interaction, the strong and the weak interaction and gravity. The Standard Model is a theory based on the principles of quantum mechanics and is able to describe three of the four forces. The only interaction that cannot be described in this framework is gravity.
In all of the interactions described by the Standard Model the matter particles are given by Spin 1/2 fermions and the interactions between them are described by the exchange of a Spin 1 boson.
In the Standard Model the freedom what one can write in the Lagrangian is restricted by the requirement of gauge invariance. More precisely, the gauge group of the Standard Model is given by a $SU(3) \times SU(2) \times U(1)$ symmetry, where the $SU(3)$ of the strong interaction is unbroken, but the $SU(2) \times U(1)$ symmetry of the so called electroweak interaction is spontaneously broken to an electromagnetic $U(1)$ symmetry. The origin of this symmetry breaking is still not clear. The Higgs mechanism postulates one solution to this problem however the resulting Higgs boson has not yet been found.
Matter particles can be divided in two parts, lepton and quarks. Quarks are strong interacting particles, whereas leptons are not. However both come in three different copies, called generations.

1.2 Calculations in perturbation theory

Nowadays experiments have reached a very high level of precision. To reach the same level of precision on the theoretical side means complex and time-consuming calculations. But these precision calculations are of great importance because it is the difference between the measurement and the prediction of the Standard Model that constrains the space for new physics. So it is very important to be able to calculate the results predicted by the Standard Model to a sufficient accuracy in order to be in a position to claim whether there is new physics or not.
With the mathematical methods that are available it is not possible to calculate a quantum mechanical scattering process with arbitrary precision. The strategy physicists have been following since the invention of quantum mechanics is to treat the interaction as a small perturbation of the free Hamiltonian and perform a series expansion in powers of the coupling constant. This is possible, as long as the coupling constant is smaller than of $\mathcal{O}(1)$. Throughout this thesis we are interested in

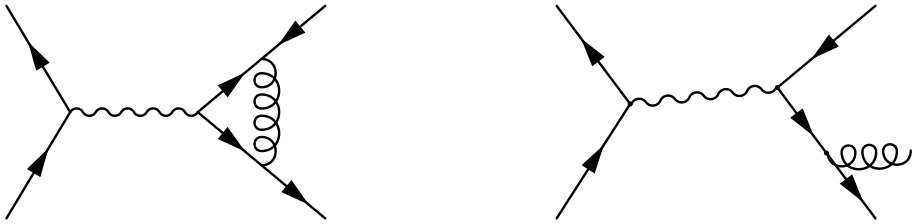


Figure 1.1: Example for corrections at next-to-leading order in QCD. The feynman diagram on the left hand side shows a loop diagram leading to a vertex correction where a gluon is exchanged between the two outgoing fermions. The diagram on the right hand side shows an example for a real emission, in this case the radiation of an additional gluon off one of the final state fermions.

calculations involving the strong interaction with its coupling constant α_s. In the energy range of colliders like Tevatron or LHC, the value of this strong coupling constant is approximately 0.1 which means that a series expansion is possible, however the series converges much slower as for the electromagnetic interaction where the coupling constant α_{QED} is of the order of 0.01. One can therfore expect, that contributions coming from higher order terms in this perturbation theory may have a sizeable contribution. The first term of this series expansion is called the leading order term (LO), the next term in the expansion is called the next-to-leading-Order term (NLO), and so on.

1.2.1 Structure of higher order calculations

The calculations of contributions to a cross section in higher order perturbation theory are divided in two parts, the virtual corrections and the real emission terms. Virtual corrections means the creation and annihilation of particles within the energy-time uncertainty principle. Therefore these particles can not be seen in a detector. What also can not be seen in a detector is the radiation of an additional particle if this particle is soft and/or collinear to another particle. A full calculation of higher order corrections are always made up of these two different contributions. An illustration of virtual and real corrections is shown in Figure 1.1

Divergencies in higher order corrections

Divergencies are an inherent property of quantum field theory. It took several decades until it was completely understood where and why these divergencies arise and how they should be treated.
One can distinguish two different types of singularities, called ultraviolet (UV) and infrared (IR) singularities. Ultraviolet singularities can arise in loop diagrams when

the loop momentum goes to infinity. They are removed from the theory by renormalization, which means a redefinition of physical parameters such as the mass or the coupling constant. If there are enough physical parameters to 'hide' all the ultraviolet divergencies in the parameters, the theory is called renormalisable [1].
Infrared singularities arise in loop diagrams as well as in real emission diagrams, when an external massless particle becomes either soft or becomes collinear to another massless particle. However it turns out that the infrared singularities of the virtual corrections exactly cancel with the infrared singularities of real emission processes. This has first been found by Bloch and Nordsieck [2] for the bremsstrahlung process when the photon energy vanishes, and has later been proven by Kinoshita, Lee and Nauenberg [3, 4] and is now known as the KLN-theorem.
Concerning the infrared singularities, there is however still an issue in practical calculations. The processes are in most cases too complex to find a closed analytic expression for the cross section and the various distributions. Especially the integration over a high dimensional phase space is then no longer possible. In these case one uses numerical methods based on Monte Carlo methods. To calculate a cross section the integration over the phase space is required. The dimension of the phase space depends on the number of final state particles and is given by

$$d = 3n - 4, \tag{1.1}$$

where n is the number of the final state particles. However this means that the dimension of the phase space for the virtual contribution is a different from the phase space of the real emission contribution because the real emission has one more final state particle. This again implies that the two integrations are done separately. But then one faces the problem that one integrates numerically over a divergent function which necessarily does not lead to a well-defined result. There are several proposals in the literature to circumvent this problem [5–13]. Independent of which method is chosen it turns out to be a challenging and time consuming part of the calculation to deal with the infrared singularities in a proper way.
A major part of this thesis is dedicated to the question how this problem can be solved in a fully automated way and for general purposes.

1.3 QCD

Quantum chromodynamics describes the quantum mechanical treatment of the strong interaction responsible for the interactions between protons and neutrons inside an atomic core as well as for the interactions of quarks inside the nucleons. In the Standard Model, QCD is described as a non-abelian gauge theory with an $SU(3)$ symmetry [14, 15]. The quarks transform as a triplet under an $SU(3)$ gauge transformation therefore coming in three different types according to the three different 'charges' of $SU(3)$. This charge, which is the equivalent to the electromagnetic charge in quantum electrodynamics is called color and labelled with 'red', 'green'

and 'blue'.
The interaction is mediated by gluons transforming as an octet under $SU(3)$, hence occurring in eight different types. There are striking differences to quantum electrodynamics. First quarks and gluons have never been observed as free particles but always bound in mesons and baryons forming a neutral bound state meaning neutral concerning the color. This behavior could so far not be derived rigorously but is a well established observation and is known as confinement.
Another observation shows that in a hadron quarks behave like free particles at high energies. This phenomenon is known as asymptotic freedom. Due to quantum corrections the strength of the strong interaction decreases with increasing energy. Although being a strange property from a classical point of view this behaviour is well understood and is only possible for a non-abelian theory. On the other hand this implies that perturbation theory is only valid for high energies (starting of order 1 GeV) because for lower energies the coupling constant is too large for a series expansion.
For colliders like Tevatron or LHC the strong coupling constant is still roughly an order of magnitude larger than the electromagnetic coupling constant. For many processes QCD corrections are therefore more important than electromagnetic corrections. This is the reason why we deal with QCD corrections in this thesis.

1.4 Status of LO and NLO calculations

Physics studies at the upcoming CERN LHC collider will frequently involve multiparticle final states. Especially searches for physics beyond the standard model rely on the reconstruction of new particles from their decay products, often through decay chains. Equally, requiring accompanying particles in the final state may serve to improve the ratio of signal to background processes, as done for example in the Higgs boson search through the vector boson fusion channel. Meaningful searches for these signals require not only a very good anticipation of the expected signal, but also of all standard model backgrounds yielding identical final state signatures. From the theoretical point of view, high precision implies that one has to go beyond the leading order in perturbation theory to be able to keep up with the precision of the measurements.
For leading order processes there have been many developments concerning event generation and simulation tools in the last two decades such as MadGraph/MadEvent [16–18] CompHEP/CalcHEP [19]/ [20], SHERPA [21] and WHIZARD [22] and also programs using different approaches such as ALPGEN [23] and HELAC [24]. All these programs are multi-purpose event generator tools, which are able to compute any process (up to technical restrictions in the multiplicity) within the standard model, or within alternative theories specified by their interaction Lagrangian or Feynman rules. They usually provide event information which can be interfaced into parton shower, hadronization and/or detector simulation.

Next-to-leading order (NLO) calculations are at present performed on a process-by-process basis. The widely-used programs MCFM [25, 26], NLOJET++ [27], MC@NLO [28, 29] and programs based on the POWHEG method [30–35] collect a variety of different processes in a standardized framework, the latter two methods also match the NLO calculation onto a parton shower.
The NLO QCD corrections to a given process with a n-parton final state receive two types of contributions: the one-loop virtual correction to the $(2 \to n)$-parton scattering process, and the real emission correction from all possible $(2 \to n+1)$-parton scattering processes. For the numerical evaluation, one has to be able to compute both types of contributions separately.
The computation of one-loop corrections to multi-particle scattering amplitudes was performed on a case-by-case basis up-to-now, the calculational complexity increased considerably with increasing number of external partons. Since only a limited number of one-loop integrals can appear [36, 37] in the final result, the calculation of one-loop corrections can be reformulated as determination of the coefficients of these basis integrals, plus potential rational terms. Enormous progress [38–57] has been made in the recent past in the systematic determination of the one-loop integral coefficients and rational terms, and steps towards fully automated programs for the calculation of one-loop multi-parton amplitudes were made with the packages CutTools [58], BlackHat [59], Rocket [60] and GOLEM [61].
Using these packages the following processes were computed recently: $t\bar{t}b\bar{b}$ [62], $H+2j$ [63], $W+3j$ [64–69], $WW+j$ [70], $t\bar{t}Z$ [71,72] and corrections to three gauge bosons [73].

1.5 Outline

This thesis is organized as follows. In chapter two we describe the so-called dipole subtraction method [12,13] as one solution to the problem dealing with infrared singularities as described above. We distinguish between the massless and the massive case and treat them separately. Chapter three describes in detail the automation and implementation of this method into the MadGraph/MadEvent package [16–18]. The result of this implementation has been published in [74] and is publicly available under the name MadDipole. Besides the implementation we perform various checks to ensure the correctness of our results.
As we will show in the following chapters it would also be convenient to have an automated tool for the integrated dipoles, where the integration over the phase space of the unresolved particle is done analytically. Although the results for the integrated dipoles are already known in the literature for the case where the integration is performed over the full phase space new results had to be derived for the possibility where parts of the phase space are cut away by a cut parameter described in chapter three. The integrated dipoles and the restrictions concerning the phase space integration are described in chapter four. Besides the implementation as a

part of the already existing MadDipole package we discuss checks that have been made to control the implementation as well as the independence of the final result of the cut parameter.
Chapter 5 contains a short manual about the MadDipole package where we explain how to install and how to use it.

In chapter six we present as a first apllication the calculation of next-to-leading order QCD corrections to the production of four final state b-quarks at the LHC. Within this thesis we restrict ourselves to the case where we only have quarks in the initial state. Besides details of the calculation we present results for the scale dependence of the NLO result as well as various distributions for important observables.
A major part of the work was the development and debugging of the necessary software tools. This has lead to numerous additional subroutines that became part of the MadGraph framework and built the MadDipole package as an add-on to the conventional MadGraph package. In appendix A we provide a short description how MadGraph and also MadDipole works. This enables us to see at which points modifications had to be made. Moreover we discuss the new developed routines and explain their function.

Chapter 2

The Dipole Subtraction formalism

This chapter explains the basic principles of the dipole formalism, developed in [12] (CS) for massless particles and later generalised to massive particles [13] (CDST).
As already explained in the introduction, infrared singularities occur in both virtual and real corrections. Although they cancel each other, in a numerical calculation the two parts are calculated separately due to the different number of final state particles which means one faces the singular terms in each part of the calculation. Therefore a formalism is needed that is able to remove the singular terms of the calculation such that a finite result remains.
There are two different types of methods, either based on phase space slicing or on subtraction.
Phase space slicing discriminates between the singular and the non-singular regions in phase space. Only in the non-singular region the full matrix element is used whereas in the singular region soft-gluon and leading-pole approximations are used such that the integration is possible [75–77].
Subtraction method in general means to find an expression that has the same singular behaviour as the original function so that the difference of these two terms is integrable. This method has first been applied in [78].
The difference between the two methods, phase space slicing and subtraction method is nicely explained in [6].
Today there are basically three types of subtraction methods used. The first one that was developed is the FKS subtraction scheme [7] followed by the dipole subtraction formalism [12, 13] and the antenna formalism [8–10, 79–81]
In this chapter we will only describe the dipole formalism as this is the method we choose for our implementation. The reasons for this choice are that it is a widely used scheme and the calculation of the color factors is easier for dipoles in the MadGraph framework as a dipole can be associated with one Feynman diagram for which routines for color calculation already exist. Also this method has been developed including all possible cases, in particular the inclusion of massive particles which so far has not been the fully case for the antenna formalism. Recent progress there can be found in [79].

2.1 Conceptual idea of dipole formalism

In this section we describe the principles of the dipole formalism as derived in [12] for the massless case and in [13] for massive particles.
If one calculates a cross section at next-to-leading order (NLO) one can write the cross section as a sum of real emission terms and virtual corrections,

$$\sigma^{NLO} \equiv \int d\sigma^{NLO} = \int_{m+1} d\sigma^R + \int_m d\sigma^V \ . \tag{2.1}$$

The basic idea of the dipole formalism is to find an expression $d\sigma^A$ that has the same infrared singularities as the real emission matrix element such that after subtracting this expression, the difference is finite. This means

$$d\sigma^{NLO} = \left[d\sigma^R - d\sigma^A\right] + d\sigma^A + d\sigma^V \ , \tag{2.2}$$

or, including the integration over phase space,

$$\sigma^{NLO} = \int_{m+1} \left[d\sigma^R - d\sigma^A\right] + \int_{m+1} d\sigma^A + \int_m d\sigma^V \ . \tag{2.3}$$

The expression $d\sigma^A$ acts as a local counterterm which means the subtraction takes place at every single phase space point and not only after integration over the phase space.
Important properties of the dipoles are the correct behaviour in the soft and collinear limit. In the soft limit the behaviour of the matrix element is known and as has been shown in [82] takes the form

$$\begin{aligned} &{}_{m+1}\langle 1,....,m+1||1,....,m+1\rangle_{m+1} \to \\ &-\frac{1}{\lambda^2}\ 4\pi\mu^{2\epsilon}\alpha_{\mathrm{S}}\ {}_m\langle 1,...,m+1|\left[\boldsymbol{J}^\mu(q)\right]^\dagger \boldsymbol{J}_\mu(q)|1,....,m+1\rangle_m \ , \end{aligned} \tag{2.4}$$

when a final state momentum p^μ vanishes, i.e.

$$p^\mu = \lambda q^\mu \ , \quad \lambda \to 0 \ . \tag{2.5}$$

The term $\boldsymbol{J}^\mu$ is the eikonal current for a soft gluon, given by

$$\boldsymbol{J}^\mu(q) = \sum_i \boldsymbol{T}_i \frac{p_i^\mu}{p_i \cdot q} \ . \tag{2.6}$$

Also the behaviour of the matrix element in the collinear limit is known. First we have to define the collinear limit. If p^μ denotes the collinear direction and $k_\perp$ the transverse direction then the collinear limit of two momenta p_i and p_j is defined as

$$p_i^\mu = zp^\mu + k_\perp^\mu - \frac{k_\perp^2}{z}\frac{n^\mu}{2pn} \ , \quad p_j^\mu = (1-z)p^\mu - k_\perp^\mu - \frac{k_\perp^2}{1-z}\frac{n^\mu}{2pn} \ . \tag{2.7}$$

The vector n^ν is an arbitrary lightlike vector which is necessary to specify the directions. It fulfills the following properties:

$$k_\perp p \;=\; 0 \;, \qquad k_\perp n = 0 \;. \tag{2.8}$$

In the limit of small $k_\perp$ one finds

$$2p_i p_j = -\frac{k_\perp^2}{z(1-z)} \;, \quad \text{for } k_\perp \to 0 \;. \tag{2.9}$$

In this limit the matrix element takes the approximate form

$$\begin{aligned} &{}_{m+1}\langle 1,....,m+1||1,....,m+1\rangle_{m+1} \to \\ &\frac{1}{p_i p_j}\, 4\pi\mu^{2\epsilon}\alpha_{\rm S}\; {}_m\langle 1,...,m+1|\; \hat{P}_{(ij),i}(z,k_\perp;\epsilon)\; |1,....,m+1\rangle_m \;\;, \end{aligned} \tag{2.10}$$

as shown in [83]. The factor $\hat{P}_{(ij),i}(z,k_\perp;\epsilon)$ is the Altarelli-Parisi splitting function in d dimensions.

To combine the subtraction terms also with the virtual corrections, the $(m+1)$-particle phase space hast to be split into an m-particle phase space and a 1-particle phase space. The total result then reads as

$$\sigma^{NLO} = \int_{m+1} \left[\left(d\sigma^R\right) - \left(d\sigma^A\right) \right] + \int_m \left[d\sigma^V + \int_1 d\sigma^A \right] \;. \tag{2.11}$$

The terms contributing to $d\sigma^A$ are now the dipoles whereas the integration over the 1-particle phase space leads to the integrated dipoles.

2.2 Dipoles in the massless case

In this section we treat all quarks as massless particles. This simplifies the formulae considerably. In the next section we then discuss the massive case. As the following is discussed in great detail in [12, 13] we will only repeat the most important facts.

2.2.1 Final-final

We start with considering only final state particles. This is for instance the case when the initial state particles are leptons. This situation is illustrated in Fig. 2.1. The particle j is the particle that becomes soft and/or collinear to the particle i. For this reason particle i is called emitter. As particle k is not directly involved in the splitting it is called spectator. The idea of constructing the dipole out of

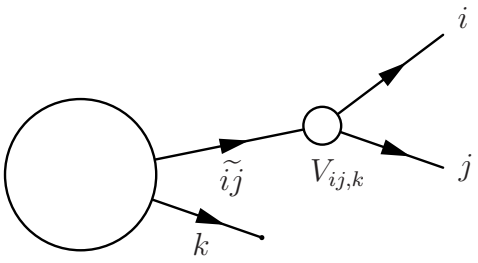


Figure 2.1: Diagram for a final-final dipole with emitter i and spectator k. j labels the particle that gets unresolved.

this diagram is as follows. If j gets unresolved one is left with a diagram with only two final state particles. An experiment would only register the combination of particle i and j. We label this combined particle with $\widetilde{ij}$. We can now build a dipole by taking such a reduced matrix element with two particles in the final state and add important properties that would get lost by just removing the unresolved particle. These important properties is first of all a propagator for the fermion line labelled with $\widetilde{ij}$. Furthermore there are additional color factors as we deal with QCD particles and there are kinematical factors which we label with $V_{ij,k}$. This object is called the splitting function. The dipole for this case is then given by (Eq. 5.2 CS)

$$\mathcal{D}_{ij,k}\left(p_1,...,p_{m+1}\right) = -\frac{1}{2p_i\cdot p_j} \tag{2.12}$$
$$\cdot\, {}_m\!<1,..,\widetilde{ij},..,\widetilde{k},..,m+1|\,\frac{\boldsymbol{T}_k\cdot\boldsymbol{T}_{ij}}{\boldsymbol{T}_{ij}^2}\,\boldsymbol{V}_{ij,k}\,|1,..,\widetilde{ij},..,\widetilde{k},..,m+1>_m\ .$$

Combining particles i and j to a new particle $\widetilde{ij}$ requires to combine the two original momenta to one. There are two requirements that have to be fulfilled: The original particles were assumed to be free final states therefore they are on-shell particles. This must then also hold for the combined particle. The second requirement is the conservation of energy and momentum. To fulfill these requirements an additional particle, the spectator, is needed. Its momentum also needs to be rescaled. The two conditions can be written as

$$\begin{aligned} p_i^2 \;&=\; p_j^2 \;=\; p_k^2 \;=\; \tilde{p}_{ij}^2 \;=\; \tilde{p}_k^2 \;=\; 0 \\ p_i^\mu \;&+\; p_j^\mu \;+\; p_k^\mu \;=\; \tilde{p}_{ij}^\mu + \tilde{p}_k^\mu\ . \end{aligned} \tag{2.13}$$

These conditions are fulfilled by the following transformation (Eq.5.3 CS)

$$\widetilde{p}_k^\mu = \frac{1}{1-y_{ij,k}}\,p_k^\mu\ \ ,\quad \widetilde{p}_{ij}^\mu = p_i^\mu + p_j^\mu - \frac{y_{ij,k}}{1-y_{ij,k}}\,p_k^\mu\ , \tag{2.14}$$

with

$$y_{ij,k} = \frac{p_ip_j}{p_ip_j + p_jp_k + p_kp_i}\ . \tag{2.15}$$

As can easily be checked, this transformation fulfills the conditions in (2.13). This does not necessarily imply that this is the only possible solution but it is one possibility.
For the actual form of the splitting function $V_{ij,k}$ we refer to the CS paper [12], for the final-final case given by Eqs.(5.29)-(5.31).

Phase space for final-final dipoles

As already mentioned at the beginning of this chapter for the integrated dipoles it is necessary to split the $(m+1)$-particle phase space into a m-particle phase space and a 1-particle phase space. The phase space spanned by the momenta p_i, p_j and p_k can be expressed with the new rescaled momenta $\tilde{p}_{ij}$ and $\tilde{p}_k$ and the momentum p_i (Eq.5.17. CS):

$$d\phi(p_i, p_j, p_k; Q) = d\phi(\widetilde{p}_{ij}, \widetilde{p}_k; Q) \ [dp_i(\widetilde{p}_{ij}, \widetilde{p}_k)] \ . \tag{2.16}$$

The 1-particle phase space can now be expressed as the integration over two kinematic variables and an angular integration. Performing this integration in d-dimensions this phase space is given by Eq.(5.20) CS,

$$\begin{aligned} [dp_i(\widetilde{p}_{ij}, \widetilde{p}_k)] &= \frac{(2\widetilde{p}_{ij}\widetilde{p}_k)^{1-\epsilon}}{16\pi^2} \frac{d\Omega^{(d-3)}}{(2\pi)^{1-2\epsilon}} \, d\tilde{z}_i \, dy_{ij,k} \, \Theta(\tilde{z}_i(1-\tilde{z}_i)) \, \Theta(y_{ij,k}(1-y_{ij,k})) \\ &\cdot \quad (\tilde{z}_i(1-\tilde{z}_i))^{-\epsilon} \, (1-y_{ij,k})^{1-2\epsilon} \, y_{ij,k}^{-\epsilon} \, , \end{aligned} \tag{2.17}$$

where the variables $\tilde{z}_i$ and $\tilde{z}_j$ are defined as

$$\tilde{z}_i = \frac{p_i p_k}{p_j p_k + p_i p_k} = \frac{p_i \widetilde{p}_k}{\widetilde{p}_{ij} \widetilde{p}_k} \, , \quad \tilde{z}_j = \frac{p_j p_k}{p_j p_k + p_i p_k} = \frac{p_j \widetilde{p}_k}{\widetilde{p}_{ij} \widetilde{p}_k} = 1 - \tilde{z}_i \, . \tag{2.18}$$

$\mathrm{d}^{d-3}\Omega$ is an element of solid angle in d-dimensions. Its integration leads to

$$\int \mathrm{d}^{d-3}\Omega = \frac{2\pi}{\pi^\epsilon \Gamma(1-\epsilon)}. \tag{2.19}$$

2.2.2 Final-initial

Many interesting processes involve hadrons particles also in the initial state. One possibility is that the emitter is a final state particle but the spectator is in the initial state. This is illustrated in Fig. 2.2. All the ideas for the final-final case also hold for the final-initial case. The dipoles are given by Eq. (5.36) CS

$$\begin{aligned} \mathcal{D}_{ij}^a(p_1, ..., p_{m+1}; p_a, ..) &= -\frac{1}{2p_i \cdot p_j} \, \frac{1}{x_{ij,a}} \\ \cdot \ {}_{m,a.}\langle 1, .., \widetilde{ij}, .., m+1; \widetilde{a}, ..| \, &\frac{\boldsymbol{T}_a \cdot \boldsymbol{T}_{ij}}{\boldsymbol{T}_{ij}^2} \, \boldsymbol{V}_{ij}^a \, |1, .., \widetilde{ij}, .., m+1; \widetilde{a}, ...\rangle_{m,a..} \ . \end{aligned} \tag{2.20}$$

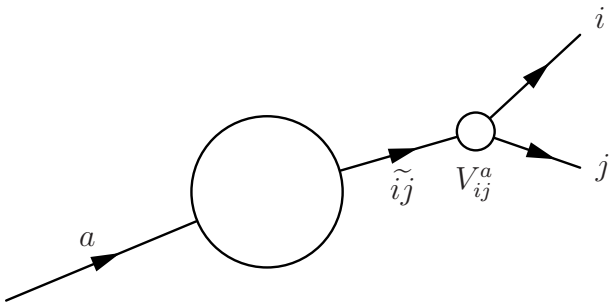


Figure 2.2: Diagram for a final-initial dipole with emitter i and unresolved particle j. In the case where the spectator is in the initial state it is labelled with a.

The new momenta $\tilde{p}_a^\mu$ and $\tilde{p}_{ij}^\mu$ again have to fulfill the on-shellness condition and preserve energy and momentum. This is assured by the transformation according to Eqs. (5.37)-(5.38) CS

$$\widetilde{p}_a^\mu = x_{ij,a}\, p_a^\mu \ , \quad \widetilde{p}_{ij}^\mu = p_i^\mu + p_j^\mu - (1 - x_{ij,a})\, p_a^\mu \ , \tag{2.21}$$

$$x_{ij,a} = \frac{p_i p_a + p_j p_a - p_i p_j}{(p_i + p_j) p_a} \ . \tag{2.22}$$

Phase space for final-initial dipoles

The phase space of the particle with final state momenta p_i, p_j and initial state momentum p_a can be rewritten in terms of the new remapped momenta by the convolution given in Eq. (5.46) CS

$$d\phi(p_i, p_j; Q + p_a) = \int_0^1 dx\ d\phi(\widetilde{p}_{ij}; Q + x p_a)\ \ [dp_i(\widetilde{p}_{ij}; p_a, x)] \ . \tag{2.23}$$

The 1-particle phase space can then be written as

$$\begin{aligned} [dp_i(\widetilde{p}_{ij}; p_a, x)] \ &= \ \frac{(2\widetilde{p}_{ij} p_a)^{1-\epsilon}}{16\pi^2}\, \frac{d\Omega^{(d-3)}}{(2\pi)^{1-2\epsilon}}\, d\tilde{z}_i\, dx_{ij,a}\, \Theta(\tilde{z}_i(1-\tilde{z}_i))\, \Theta(x(1-x)) \\ &\cdot \quad (\tilde{z}_i(1-\tilde{z}_i))^{-\epsilon}\ \delta(x - x_{ij,a})\ (1-x)^{-\epsilon} \ . \end{aligned} \tag{2.24}$$

2.2.3 Initial-final

Another possibility is that the emitter is an initial state particle while the spectator is in the final state. This situation is shown in Figure 2.3. This is very similar to the final-initial case discussed before, just the roles of emitter and spectator are interchanged. The dipoles are therefore given by Eq.(5.61) CS

$$\begin{aligned} \mathcal{D}_k^{ai}(p_1, ..., p_{m+1}; p_a) &= -\frac{1}{2 p_a \cdot p_i}\ \frac{1}{x_{ik,a}} \\ &\cdot_{m,a}\langle 1, .., \widetilde{k}, ., m+1; \widetilde{ai}|\ \frac{\boldsymbol{T}_k \cdot \boldsymbol{T}_{ai}}{\boldsymbol{T}_{ai}^2}\ \boldsymbol{V}_k^{ai}\ |1, .., \widetilde{k}, .., m+1; \widetilde{ai}\rangle_{m,a} \ . \end{aligned} \tag{2.25}$$

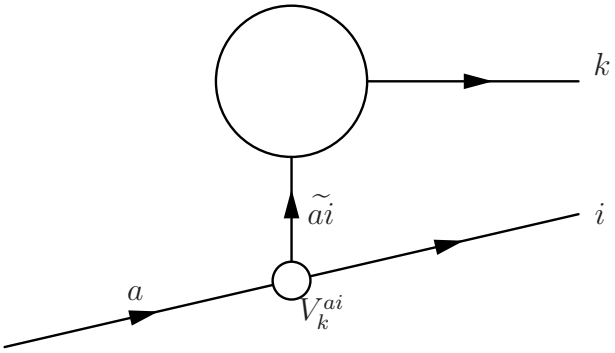


Figure 2.3: Diagram for an initial-final dipole. The emitter is labelled with a and the unresolved particle with i. The spectator is again labelled with k.

For the transformation of the momenta one finds then

$$\widetilde{p}_{ai}^{\mu} = x_{ik,a}\, p_a^{\mu} \;, \qquad \widetilde{p}_k^{\mu} = p_k^{\mu} + p_i^{\mu} - (1 - x_{ik,a})\, p_a^{\mu} \;, \tag{2.26}$$

with

$$x_{ik,a} = \frac{p_k p_a + p_i p_a - p_i p_k}{(p_k + p_i) p_a} \;. \tag{2.27}$$

Phase space for initial-final dipoles

With the definition

$$u_i = \frac{p_i p_a}{p_i p_a + p_k p_a} \tag{2.28}$$

and with the following form for the three parton phase space (Eq.(5.70) CS)

$$d\phi(p_i, p_k; Q + p_a) = \int_0^1 dx\; d\phi(\widetilde{p}_k; Q + x p_a)\;\; [dp_i(\widetilde{p}_k; p_a, x)] \;, \tag{2.29}$$

the 1-particle phase space of the momentum p_i can then be written as (Eq.(5.72) CS)

$$\begin{aligned} [dp_i(\widetilde{p}_k; p_a, x)] \;\; &= \;\; \frac{(2\widetilde{p}_k p_a)^{1-\epsilon}}{16\pi^2}\; \frac{d\Omega^{(d-3)}}{(2\pi)^{1-2\epsilon}}\; du_i\; dx_{ik,a}\; \Theta(u_i(1-u_i))\; \Theta(x(1-x)) \\ &\cdot \;\; (u_i(1-u_i))^{-\epsilon}\; \delta(x - x_{ik,a})\; (1-x)^{-\epsilon} \;. \end{aligned} \tag{2.30}$$

2.2.4 Initial-initial

The last possibility of arranging emitter and spectator is the case when both emitter and spectator are in the initial state, as shown in 2.4. The dipole has the same

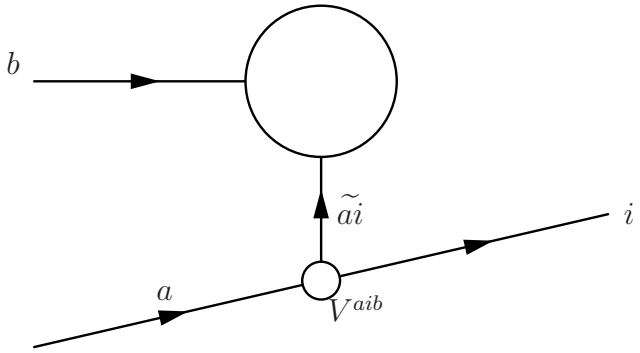


Figure 2.4: Diagram for an initial-initial dipole. The emitter is again labelled with a and the unresolved particle with i. The initial state spectator is labelled with b.

structure as for the previous cases involving initial state partons. According to Eq.(5.136) CS it is

$$\mathcal{D}^{ai,b}(p_1,...,p_{m+1};p_a,p_b) = -\frac{1}{2p_a\cdot p_i}\,\frac{1}{x_{i,ab}} \\ {}_{m,ab}\langle\widetilde{1},..,\widetilde{m+1};\widetilde{ai},b|\,\frac{\boldsymbol{T}_b\cdot\boldsymbol{T}_{ai}}{\boldsymbol{T}_{ai}^2}\,\boldsymbol{V}^{ai,b}\,|\widetilde{1},..,\widetilde{m+1};\widetilde{ai},b\rangle_{m,ab}\ . \tag{2.31}$$

Concerning the transformation of the momenta there is now a difference to the previous cases. In the cases where there is either emitter or spectator in the initial state and the other in the final state, the initial state particle gets rescaled by a factor of x which takes into account that the energy available for the actual hard scattering process is reduced due to the emission of an additional particle. The remapping of the final state particles compensates for this. This is however not possible if both particles are in the initial state. Therefore only the emitter gets rescaled and the spectator remains unchanged. To compensate for the rescaling of the emitter one applies a Lorentz transformation to all final state particles. In particular this transformation is also applied to non-QCD particles. The emitter gets rescaled by (Eq.(5.137)-(5.138) CS)

$$\widetilde{p}_{ai}^{\,\mu} = x_{i,ab}\,p_a^\mu\ , \qquad x_{i,ab} = \frac{p_ap_b - p_ip_a - p_ip_b}{p_ap_b}\ . \tag{2.32}$$

All the final state momenta k_j^μ are transformed according to (Eq.(5.139)-(5.140) CS)

$$\widetilde{k}_j^\mu = k_j^\mu - \frac{2k_j\cdot(K+\widetilde{K})}{(K+\widetilde{K})^2}\,(K+\widetilde{K})^\mu + \frac{2k_j\cdot K}{K^2}\,\widetilde{K}^\mu\ , \tag{2.33}$$

with

$$\begin{aligned} K^\mu &= p_a^\mu + p_b^\mu - p_i^\mu\ , \\ \widetilde{K}^\mu &= \widetilde{p}_{ai}^{\,\mu} + p_b^\mu\ . \end{aligned} \tag{2.34}$$

Phase space for initial-initial dipole

The phase space for this case is written as the convolution (Eq.(5.149) CS)

$$d\phi(p_i,k_1,...;p_a+p_b)=\int_0^1 dx\; d\phi(\widetilde{k}_1,...;xp_a+p_b)\;\;[dp_i(p_a,p_b,x)]\;. \tag{2.35}$$

With the help of the definition

$$\tilde{v}_i=\frac{p_ap_i}{p_ap_b}\;, \tag{2.36}$$

the 1-particle phase space $[dp_i(p_a,p_b,x)]$ can be expressed as (Eq.(5.151) CS)

$$\begin{aligned}[dp_i(p_a,p_b,x)] &= \frac{(2p_ap_b)^{1-\epsilon}}{16\pi^2}\,\frac{d\Omega^{(d-3)}}{(2\pi)^{1-2\epsilon}}\,d\tilde{v}_i\,dx_{i,ab}\,\Theta(x(1-x))\,\Theta(\tilde{v}_i)\,\Theta\!\left(1-\frac{\tilde{v}_i}{1-x}\right)\\ &\cdot\quad (1-x)^{-2\epsilon}\;\delta(x-x_{i,ab})\;\left[\frac{\tilde{v}_i}{1-x}\left(1-\frac{\tilde{v}_i}{1-x}\right)\right]^{-\epsilon}\;. \end{aligned}\tag{2.37}$$

2.3 Massive dipoles

In this section we discuss the generalisation of the dipole formalism now also allowing massive particles in the final state. Initial state particles are always assumed to be massless. The detailed analysis can be found in [13].

2.3.1 Final-final

To prevent more confusion than absolutely necessary we follow the notation introduced in the literature meaning that the unresolved particle is now labelled with i instead of j as in the massless case.
The only modification of the dipole compared to the massless case is that the propagator is no longer massless. Therefore we can immediately write (Eq.(5.2) CDST)

$$\begin{aligned}&\mathcal{D}_{ij,k}(p_1,\ldots,p_{m+1})=\\ &-\frac{1}{(p_i+p_j)^2-m_{ij}^2}\;{}_m\langle\ldots,\widetilde{ij},\ldots,\tilde{k},\ldots|\frac{\boldsymbol{T}_k\cdot\boldsymbol{T}_{ij}}{\boldsymbol{T}_{ij}^2}\mathbf{V}_{ij,k}|\ldots,\widetilde{ij},\ldots,\tilde{k},\ldots\rangle_m\end{aligned}\tag{2.38}$$

Introducing the triangular function $\lambda(x,y,z)$ with

$$\lambda(x,y,z)=x^2+y^2+z^2-2xy-2xz-2yz\;, \tag{2.39}$$

the transformation of the momenta is given by (Eq.(5.9) CDST)

$$\begin{aligned}\widetilde{p}_k^{\,\mu}&=\frac{\sqrt{\lambda(Q^2,m_{ij}^2,m_k^2)}}{\sqrt{\lambda(Q^2,(p_i+p_j)^2,m_k^2)}}\left(p_k^\mu-\frac{Qp_k}{Q^2}Q^\mu\right)+\frac{Q^2+m_k^2-m_{ij}^2}{2Q^2}Q^\mu\;,\\ \widetilde{p}_{ij}^{\,\mu}&=Q^\mu-\widetilde{p}_k^{\,\mu}\;,\end{aligned}\tag{2.40}$$

where Q^μ is the sum of the three momenta, or, equivalent

$$Q^\mu = \widetilde{p}_{ij}^{\,\mu} + \widetilde{p}_k^{\,\mu} \,. \tag{2.41}$$

Massive phase space for final-final dipoles

We start considering the massive phase space by the definition of the rescaled masses

$$\mu_n = \frac{m_n}{\sqrt{Q^2}} \,. \tag{2.42}$$

The three particle phase space generalises to (Eq.(5.10) CDST)

$$\mathrm{d}\phi(p_i,p_j,p_k;Q) = \mathrm{d}\phi(\widetilde{p}_{ij},\widetilde{p}_k;Q)\,[\mathrm{d}p_i(\widetilde{p}_{ij},\widetilde{p}_k)]\,\Theta(1-\mu_i-\mu_j-\mu_k)\,. \tag{2.43}$$

The integral over the 1-particle phase space can be found in Eq.(5.11).CDST

$$\begin{aligned}\int[\mathrm{d}p_i(\widetilde{p}_{ij},\widetilde{p}_k)] = {}& \frac{1}{4}(2\pi)^{-3+2\epsilon}(Q^2)^{1-\epsilon}(1-\mu_i^2-\mu_j^2-\mu_k^2)^{2-2\epsilon}\left[\lambda(1,\mu_{ij}^2,\mu_k^2)\right]^{\frac{-1+2\epsilon}{2}}\int \mathrm{d}^{d-3}\Omega \\ & \times\int_{y_-}^{y_+}\mathrm{d}y_{ij,k}\,(1-y_{ij,k})^{1-2\epsilon}\left[\mu_i^2+\mu_j^2+(1-\mu_i^2-\mu_j^2-\mu_k^2)y_{ij,k}\right]^{-\epsilon} \\ & \times\int_{z_-(y_{ij,k})}^{z_+(y_{ij,k})}\mathrm{d}\tilde{z}_i\,\left[z_+(y_{ij,k})-\tilde{z}_i\right]^{-\epsilon}\left[\tilde{z}_i-z_-(y_{ij,k})\right]^{-\epsilon}\,.\end{aligned} \tag{2.44}$$

The variables $\tilde{z}_{i,j}$ and $y_{ij,k}$ are defined as in the massless case. The integration boundaries of the two integration variables depend on the masses and are given by

$$\begin{aligned} y_- &= \frac{2\mu_i\mu_j}{1-\mu_i^2-\mu_j^2-\mu_k^2}\,, \qquad y_+ = 1-\frac{2\mu_k(1-\mu_k)}{1-\mu_i^2-\mu_j^2-\mu_k^2}\,, \\ z_\pm(y_{ij,k}) &= \frac{2\mu_i^2+(1-\mu_i^2-\mu_j^2-\mu_k^2)y_{ij,k}}{2[\mu_i^2+\mu_j^2+(1-\mu_i^2-\mu_j^2-\mu_k^2)y_{ij,k}]}(1\pm v_{ij,i}v_{ij,k})\,, \end{aligned} \tag{2.45}$$

with

$$\begin{aligned} v_{ij,k} &= \frac{\sqrt{[2\mu_k^2+(1-\mu_i^2-\mu_j^2-\mu_k^2)(1-y_{ij,k})]^2-4\mu_k^2}}{(1-\mu_i^2-\mu_j^2-\mu_k^2)(1-y_{ij,k})}, \\ v_{ij,i} &= \frac{\sqrt{(1-\mu_i^2-\mu_j^2-\mu_k^2)^2y_{ij,k}^2-4\mu_i^2\mu_j^2}}{(1-\mu_i^2-\mu_j^2-\mu_k^2)y_{ij,k}+2\mu_i^2}\,, \end{aligned} \tag{2.46}$$

(Eq.5.13-5.14 CDST).

2.3.2 Final-initial

Also for the final-initial case, the only difference to the massless dipole is taking into account a massive propagator instead of a massless (Eq.(5.40) CDST):

$$\mathcal{D}^a_{ij}(p_1,\dots,p_{m+1};p_a,\dots) = \\ -\frac{1}{(p_i+p_j)^2-m_{ij}^2}\,\frac{1}{x_{ij,a}}\;{}_{m,a}\langle\dots,\widetilde{ij},\dots;\tilde{a},\dots|\frac{\boldsymbol{T}_a\cdot\boldsymbol{T}_{ij}}{\boldsymbol{T}_{ij}^2}\mathbf{V}^a_{ij}|\dots,\widetilde{ij},\dots;\tilde{a},\dots\rangle_{m,a}\,. \quad (2.47)$$

The structure of the transformation of the momenta remains unchanged only the variable $x_{ij,a}$ contains mass dependencies (Eq.(5.42)-(5.43) CDST):

$$\begin{aligned} x_{ij,a} &= \frac{p_ap_i+p_ap_j-p_ip_j+\frac{1}{2}(m_{ij}^2-m_i^2-m_j^2)}{p_ap_i+p_ap_j}, \\ \tilde{p}^\mu_a = x_{ij,a}p^\mu_a, &\qquad \widetilde{p}^\mu_{ij} = p^\mu_i+p^\mu_j-(1-x_{ij,a})p^\mu_a. \end{aligned} \quad (2.48)$$

Phase space for massive final-initial dipoles

The phase space is described in Eqs.(5.46)-(5.49) CDST. The three particle phase space including massive particles is given by

$$\mathrm{d}\phi(p_i,p_j;Q+p_a) = \int_0^1 \mathrm{d}x\;\mathrm{d}\phi\Big(\widetilde{p}_{ij};Q+xp_a\Big)\;[\mathrm{d}p_i(\widetilde{p}_{ij};p_a,x)]\;\Theta(x_+-x)\;, \quad (2.49)$$

with $Q=\widetilde{p}_{ij}-\tilde{p}_a$. The difference to the massless case is given by the fact that due to the masses there is an upper limit of the variable x, given by

$$x_+ = 1+\mu_{ij}^2-(\mu_i+\mu_j)^2\;. \quad (2.50)$$

The 1-particle phase space is then

$$\int[\mathrm{d}p_i(\widetilde{p}_{ij};p_a,x)] = \frac{1}{4}(2\pi)^{-3+2\epsilon}(2\widetilde{p}_{ij}p_a)^{1-\epsilon}\int_0^{x_+}\mathrm{d}x_{ij,a}\,\delta(x-x_{ij,a})\,(1-x+\mu_{ij}^2)^{-\epsilon} \\ \times\int\mathrm{d}^{d-3}\Omega\int_{z_-(x)}^{z_+(x)}\mathrm{d}\tilde{z}_i\;[z_+(x)-\tilde{z}_i]^{-\epsilon}\,[\tilde{z}_i-z_-(x)]^{-\epsilon}\;, \quad (2.51)$$

where the limits on the z-integration are

$$z_\pm(x) = \frac{1-x+\mu_{ij}^2+\mu_i^2-\mu_j^2\pm\sqrt{(1-x+\mu_{ij}^2-\mu_i^2-\mu_j^2)^2-4\mu_i^2\mu_j^2}}{2(1-x+\mu_{ij}^2)}, \quad (2.52)$$

and the rescaled masses have been defined as

$$\mu_n = \frac{m_n}{\sqrt{2\widetilde{p}_{ij}p_a}}\;. \quad (2.53)$$

2.3.3 Initial-final

Keeping in mind that we treat all initial state particles massless, the dipole takes the same form as in the massless case:

$$\mathcal{D}_j^{ai}(p_1,\dots,p_{m+1};p_a,\dots) = \\ -\frac{1}{2p_ap_i}\,\frac{1}{x_{ij,a}}\;{}_{m,\widetilde{ai}}\langle\dots,\tilde{j},\dots;\widetilde{ai},\dots|\frac{\boldsymbol{T}_j\cdot\boldsymbol{T}_{ai}}{\boldsymbol{T}_{ai}^2}\mathbf{V}_j^{ai}|\dots,\tilde{j},\dots;\widetilde{ai},\dots\rangle_{m,\widetilde{ai}}\,. \quad (2.54)$$

The same holds for the transformation of the momenta, hence

$$\tilde{p}_{ai}^{\mu} = x_{ij,a}p_a^{\mu}, \qquad \tilde{p}_j^{\mu} = p_i^{\mu}+p_j^{\mu}-(1-x_{ij,a})p_a^{\mu}, \quad (2.55)$$

$$x_{ij,a} = \frac{p_ap_i+p_ap_j-p_ip_j}{p_ap_i+p_ap_j}\,. \quad (2.56)$$

Phase space for initial-final dipoles

The rescaled mass of the spectator is now defined by Eq.(5.77) CDST

$$\mu_j = \frac{m_j}{\sqrt{2\tilde{p}_jp_a}}\,. \quad (2.57)$$

Following the derivation of Eqs.(5.76)-(5.80) CDST the phase space takes the form

$$\mathrm{d}\phi(p_i,p_j;Q+p_a) = \int_0^1 \mathrm{d}x\,\mathrm{d}\phi\Big(\tilde{p}_j;Q+xp_a\Big)[\mathrm{d}p_i(\tilde{p}_j;p_a,x)]\,, \quad (2.58)$$

and the integration over the 1-particle phase space leads to

$$\int[\mathrm{d}p_i(\tilde{p}_j;p_a,x)] = \frac{1}{4}(2\pi)^{-3+2\epsilon}(2\tilde{p}_jp_a)^{1-\epsilon}\int_0^1\mathrm{d}x_{ij,a}\,\delta(x-x_{ij,a})\,(1-x+\mu_j^2)^{-\epsilon} \\ \times\int\mathrm{d}^{d-3}\Omega\int_0^{z_+(x)}\mathrm{d}\tilde{z}_i\;[z_+(x)-\tilde{z}_i]^{-\epsilon}\,\tilde{z}_i^{-\epsilon}\,, \quad (2.59)$$

where Q is

$$Q = p_i+p_j-p_a = \tilde{p}_j-\tilde{p}_{ai}\,. \quad (2.60)$$

In this case there is only an upper limit on the z-integration which is

$$z_+(x) = \frac{1-x}{1-x+\mu_j^2}\,. \quad (2.61)$$

2.3.4 Initial-initial

As there are only massless particles allowed in the initial state this case remains the same as in the previous section about massless dipoles. The treatment of massive particles in the initial state for the special case of photon radiation can be found in [84].

Chapter 3

MadDipole - Automation of the Dipole formalism

3.1 MadGraph/MadEvent

In this section we describe the basic concepts of the MadGraph/MadEvent package as this has been used as the starting point for the development of the MadDipole package.
The MadGraph/MadEvent package is publicly available an can be downloaded from the website `http://madgraph.hep.uiuc.edu/`. This website can also be used to generate and run code online.
Although it comes in one package, MadGraph and MadEvent are two completely different programs with different purposes that have to be treated separately.

3.1.1 MadGraph

MadGraph is a program whose output is a Fortran code that is able to calculate the squared matrixelement for a specified process and a given phase space point [16]. At the beginning the user has to specify the process, e.g `e+ e- -> u u~` and MadGraph returns a single subroutine of the form `SMATRIX(P,ANS)`, where `P` is the phase space point that has to be provided by the user and `ANS` is the squared matrixelement which is the output of the subroutine. This result includes the sum over color and helicity as well as averaging over initial state spin and color.
MadGraph is based on a Feynman diagrammatic approach. The individual diagrams are calculated by using the HELAS routines [85]. It is worth mentioning that in spite of the Feynman diagrammatic approach the calculation is faster than one might naively expect because MadGraph is able to recycle parts of amplitudes that are the same for different diagrams without recalculating them.
The principles how the squared matrix elements are calculated are as follows. For the specified process there is a list of all possible helicity combinations for the external

particles. For each helicity combination the squared matrix element is calculated and then the contributions of the various helicity combinations are summed. This implies that the user can have access to each single helicity combination.
As several amplitudes may have the same color structure all amplitudes are calculated without the color and then ordered according to their color structure. This is done by introducing an array called `JAMP()` where each entry corresponds to one particular color structure. In general several amplitudes contribute to one entry in `JAMP()` but also one amplitude can contribute to several entries. This is especially the case for three and four gluon vertices as these can be reduced to different color structures. The array `JAMP()` is a vector in color space and the full squared matrix element is given by the multiplication of this array with a color matrix and the complex conjugate of the array, i.e.

$$|\mathcal{M}|^2 \;=\; \left(\; \texttt{JAMP(1)}^{\dagger},\;\; \cdots \;\right) \begin{pmatrix} \ddots & & \\ & \textit{color matrix} & \\ & & \ddots \end{pmatrix} \begin{pmatrix} \texttt{JAMP(1)} \\ \vdots \\ \vdots \end{pmatrix} \tag{3.1}$$

The color matrix is calculated when the code is written and is therefore already given in the subroutine. The squared matrix element in (3.1) is evaluated only for a single helicity combination. Afterwards the combinations are summed.
A major advantage of the code produced by MadGraph is that it is completely framework independent and can be used in an arbitrary environment.

3.1.2 MadEvent

Producing cross sections and distributions requires integration over the phase space. For multi leg processes these high dimensional integrations are far from trivial hence sophisticated techniques are required. This is the role MadEvent is playing [17]. Taking the code produced by MadGraph in a previous step it integrates over the phase space using Monte-Carlo methods combined with a VEGAS algorithm [86]. As described in [17] the idea is to write the function that should be integrated in terms of basis functions, i.e.

$$f = \sum_{i=1}^{n} f_i \;, \tag{3.2}$$

so the integration reduces to a sum of independent integrations. This is the basic idea of a multi-channel integration. In particular MadEvent chooses the following basis:

$$f_i = \frac{|A_i|^2}{\sum_i |A_i|^2} |A_{\mathrm{tot}}|^2 \;. \tag{3.3}$$

In this context A_i is the amplitude of a single diagram and A_{tot} is the full amplitude. From (3.3) it is clear that the f_i have the same peak structure as the squared amplitude $|A_i|^2$. Knowing the peak structure from the MadGraph output the integration

variables for each channel are chosen such that they correspond with the kinematic variables that contain a peak structure. This leads to a very efficient integration. Besides the integration MadEvent delivers an event file according to the Les Houches accord [87].

3.2 Construction of dipole terms

In this chapter we explain how the dipole formalism has been implemented in the MadGraph framework in a fully automated way. It has also been automated in the SHERPA framework [88] and the TeVJet framework [89], and recently in the form of independent libraries [90] interfaced to MadGraph and in a generalised form in HELAC [91]. The fundamental building blocks of the subtraction terms in the dipole formalism [12,13] are dipole splitting functions $\mathbf{V}_{ij,k}$, which involve only three partons: emitter i, unresolved parton j, spectator k. A dipole splitting function accounts for the collinear limit of j with i, and for parts of the soft limit of j in between i and k. The dipole factors, which constitute the subtraction terms, are obtained by multiplication with reduced matrix elements, where partons i, j and k are replaced by recombined pseudo-partons $\widetilde{ij}$, $\widetilde{k}$. The full soft behavior is recovered after summing all dipole factors.
Throughout the whole thesis we are using the notation introduced in Refs. [12] and [13]. Independent of whether we have initial or final state particles we can write an arbitrary dipole in the form

$$\mathcal{D}_{ij,k} \quad \sim \quad {}_m\langle 1,...\tilde{ij},...,\tilde{k},...,m+1|\frac{\mathbf{T}_k\cdot\mathbf{T}_{ij}}{\mathbf{T}_{ij}^2}\mathbf{V}_{ij,k}|1,...\tilde{ij},...,\tilde{k},...,m+1\rangle_m. \quad (3.4)$$

The amplitude factors $\langle\ldots|$ ('bra') and $|\ldots\rangle$ ('ket') on the right hand side are tensors in color space. The helicities of the external particles in them are a priori fixed (but can be summed over for unpolarized processes), while the helicities of the pseudo-partons have to be summed over after contraction with the dipole splitting function. These Born-level amplitude factors are provided by the usual MadGraph code. The two elements that combine the ket with the bra are the additional color structure $\frac{\mathbf{T}_k\cdot\mathbf{T}_{ij}}{\mathbf{T}_{ij}^2}$ and the dipole splitting function $\mathbf{V}_{ij,k}$.

3.3 Color and helicity management

For the calculation of the color factors there already exist routines in the MadGraph program. Our intention was to use exactly these routines because this code is very well-confirmed and efficient. We have included the additional color operator, $\mathbf{T}_k\cdot\mathbf{T}_{ij}$, by rewriting the internal MadGraph color labelling for the ket-side only. After insertion of this color operator the color structure is no longer multiplied by its own complex conjugate and therefore the routine that squares the color needed

to be altered, to multiply the modified ket by its original complex conjugate. We emphasize that due to the factorial growth of the color factors MadGraph can in general not handle more than seven colored particles.

For the insertion of the splitting function $\mathbf{V}_{ij,k}$ several changes with respect to the original code are required. One has to keep in mind that in general the splitting function is a tensor in helicity space, *i.e.*,

$$\mathbf{V}_{ij,k} \equiv \langle \mu | \mathbf{V}_{ij,k} | \nu \rangle = \mathbf{V}^{\mu\nu}_{ij,k}.$$

As MadGraph deals with helicity amplitudes, we have to write the dipole in a slightly different way to be able to include the calculation of the splitting function in the code. Neglecting the color for a moment we start from the definition of the dipole in (3.4) and by inserting a full set of helicity states $-g_{\mu\nu} = \sum_\lambda \epsilon^*_\mu(\lambda)\epsilon_\nu(\lambda)$ we get

$$\begin{aligned}
\mathcal{D}_{ij,k} \;&\sim\; {}_m\langle 1, ...\tilde{ij}, ..., \tilde{k}, ..., m+1|_\mu \mathbf{V}^{\mu\nu}_{ij,k}\ {}_\nu|1, ...\tilde{ij}, ..., \tilde{k}, ..., m+1\rangle_m \qquad (3.5)\\
&=\; {}_m\langle 1, ...\tilde{ij}, ..., \tilde{k}, ..., m+1|_{\mu'} \left(-g^{\mu'}_{\ \mu}\right) \mathbf{V}^{\mu\nu}_{ij,k} \left(-g_\nu^{\ \nu'}\right) {}_{\nu'}|1, ...\tilde{ij}, ..., \tilde{k}, ..., m+1\rangle_m\\
&=\; \sum_{\lambda_a,\lambda_b} {}_m\langle \dots|_{\mu'}\ \epsilon^{*\mu'}(\lambda_b)\epsilon_\mu(\lambda_b)\mathbf{V}^{\mu\nu}_{ij,k}\ \epsilon^*_\nu(\lambda_a)\epsilon^{\nu'}(\lambda_a)\ {}_{\nu'}|\dots\rangle_m\\
&=\; \sum_{\lambda_a,\lambda_b} {}_m\langle \dots|_{\lambda_b}\ V(\lambda_b,\lambda_a)\ {}_{\lambda_a}|\dots\rangle_m
\end{aligned}$$

with $V(\lambda_b,\lambda_a) = \epsilon_\mu(\lambda_b)\ \mathbf{V}^{\mu\nu}_{ij,k}\ \epsilon^*_\nu(\lambda_a)$ and $\epsilon^\mu(\lambda)\,{}_\mu|\dots\rangle_m = {}_\lambda|\dots\rangle_m$.
The polarization vectors are calculated using the HELAS routines [85] already available in the MadGraph code. The parts that are diagonal in helicity space are trivial to calculate in that sense that one only has to multiply the MadGraph output for the squared amplitude for a given helicity combination with the splitting function. To calculate the off-diagonal helicity terms, the amplitude for each helicity combination is stored and then combined with the corresponding amplitude with opposite helicity.
For the calculation of the splitting functions and for the remaping of the momenta we use modified versions of the routines used in MCFM [25, 26].

3.4 Massive particles

If some of the masses of the external particles are non-zero, in particular for processes involving top and/or bottom quarks, there are dipoles for which the unresolved parton is massive. In this case the collinear singularities are regulated by the mass of the unresolved parton and the unsubtracted matrix element does therefore no longer diverge in these collinear limits, but only develops potentially large logarithms. Our code still calculates all possible dipoles, also in which the unresolved parton is massive, but puts them in a separate subroutine, `dipolsumfinite(...)`, that is not

evaluated by default. In the limit of large center of mass energy or, similarly, small external masses, the user can easily include the non-divergent dipoles to subtract the associated large logarithms, which can then be included analytically through the integrated subtraction terms. In the limit of zero external masses we have checked that the results obtained after summing all dipoles are the same as obtained by generating the code with massless particles from the start.

3.5 Phase space restrictions

The calculation of the subtraction terms is only necessary in the vicinity of a soft and/or collinear limit. Away from these limits the amplitude is finite and there is in principle no need to calculate the computationally heavy subtraction terms. The distinction between regions near to a singularity from regions without need for a subtraction can be parameterized by a parameter usually labelled with α and $\alpha \in [0, 1]$, which was introduced in Ref. [92] for processes involving partons only in the final state. The case with incoming hadrons, *i.e.*, with partons in the initial state, is described in Ref. [27].
Using the notation of Ref. [27], the contribution from the subtraction term to the differential cross section can be written as

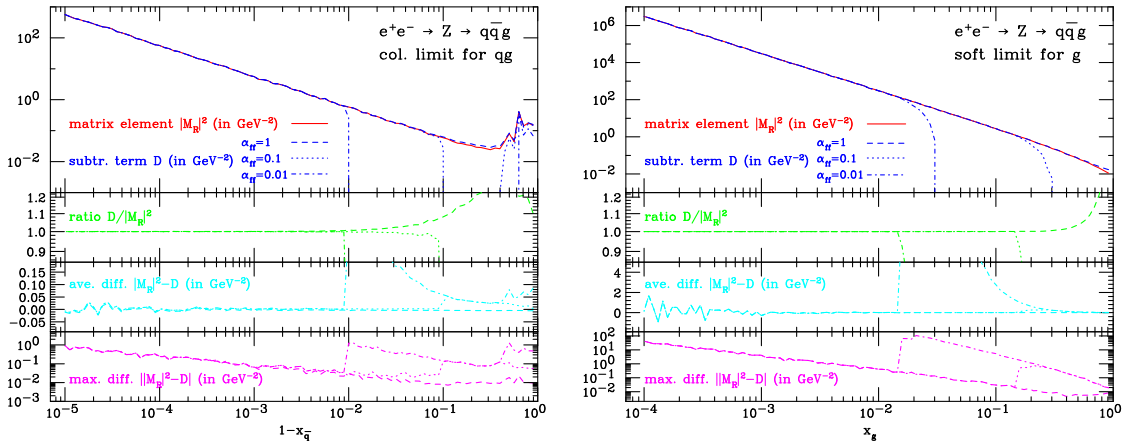
$$
\begin{aligned}
d\sigma^A_{ab} = & \sum_{\{n+1\}} d\Gamma^{(n+1)}(p_a,p_b,p_1,...,p_n+1)\frac{1}{S_{\{n+1\}}} \\
& \times\Bigg\{ \sum_{\substack{\text{pairs}\\ i,j}}\sum_{k\neq i,j} \mathcal{D}_{ij,k}(p_a,p_b,p_1,...,p_{n+1}) F_J^{(n)}(p_a,p_b,p_1,..,\tilde{p}_{ij},\tilde{p}_k,..,p_{n+1})\Theta(y_{ij,k}<\alpha) \\
& \quad + \sum_{\substack{\text{pairs}\\ i,j}} \Big[\mathcal{D}^a_{ij}(p_a,p_b,p_1,...,p_{n+1}) F_J^{(n)}(\tilde{p}_a,p_b,p_1,..,\tilde{p}_{ij},..,p_{n+1})\Theta(1-x_{ij,a}<\alpha) \\
& \qquad + (a\leftrightarrow b)\Big] \\
& \quad + \sum_{i\neq k} \Big[\mathcal{D}^{ai}_k(p_a,p_b,p_1,...,p_{n+1}) F_J^{(n)}(\tilde{p}_a,p_b,p_1,..,\tilde{p}_k,..,p_{n+1})\Theta(u_i<\alpha) + (a\leftrightarrow b)\Big] \\
& \quad + \sum_{i} \Big[\mathcal{D}^{ai,b}(p_a,p_b,p_1,...,p_{n+1}) F_J^{(n)}(\tilde{p}_a,p_b,\tilde{p}_1,...,\tilde{p}_{n+1})\Theta(\tilde{v}_i<\alpha) + (a\leftrightarrow b)\Big] \Bigg\} \ .
\end{aligned}
\tag{3.6}
$$

The functions $\mathcal{D}_{ij,k}$, $\mathcal{D}^a_{ij}$, $\mathcal{D}^{ai}_k$ and $\mathcal{D}^{ai,b}$ are the dipole terms for the various combinations for emitter and spectator. $\sum_{\{n+1\}}$ denotes the summation over all possible configurations for this $(n+1)$-particle phase space which is labelled as $d\Gamma^{(n+1)}$ and the factor $S_{\{n+1\}}$ is the symmetry factor for identical particles. We have introduced four different α-parameters, one for each type of dipoles. In our code they are called `alpha_ff`, `alpha_fi`, `alpha_if` and `alpha_ii` for the final-final, finial-initial, initial-final and initial-initial dipoles, respectively. The actual values for these parameters are by default set to unity, corresponding to the original formulation of the dipole subtraction method [12,13], but can be changed by the user in the file `dipolsum.f`. It has to be kept in mind that the integrated dipole factors, which are to be added with the virtual n-parton contribution, will also depend on α. For case of massless partons, the α-dependence of the integrated terms is stated in [27,92] while for massive partons results for most cases can be found in [93,94].

3.6 Checks

The MadDipole package provides a code, `check_dip.f`, which allows the user to test the limits of the $(n+1)$-particle matrix element and the dipole subtraction terms. This code builds up a trajectory of randomly selected phase space points approaching a given soft or collinear limit of the $(n+1)$-parton matrix element and yields the values of matrix element, sum of all the dipoles, and their ratio along this trajectory. The program uses the RAMBO generator [95] for the calculation of the phase space point. The result is printed to the screen in a small table for which each

successive row is closer to the singularity. The ratio between matrix element and the sum of the dipoles should go to unity. We have tested our code in all possible limits, both for massless as well as massive dipoles and found no inconsistencies. Choosing small values for α-parameters, *e.g.*, $\alpha = 0.1$, improves the computation time and the convergence of the subtraction procedure.
To show that the subtraction terms are implemented correctly we provide a couple of examples in the form of plots and argue that the cancellation between the matrix element squared and the subtraction term is as expected. In the figures 3.1 and 3.2 we show the matrix element squared and the subtraction term as a function of a variable that represents a soft and/or collinear limit of the process specified. For these figures we have binned the x-axis (equally sized bins for the logarithmic scale) and generated random points in phase space to fill each of the bins with exactly 100 events. In the upper plot, $|M_R|^2$ and D are the per bin averages of the matrix element squared and the subtraction term, respectively. The second to upper plot shows the per bin average of the ratio of the matrix element squared and the subtraction term, while the third plot from the top shows the per bin average of the difference. The lowest plot shows the absolute value of the maximal difference among the 100 points in a bin. To show the effects of the phase space restriction for the dipoles, see section 3.5, all the plots are given for $\alpha = 1$ (dashed lines), $\alpha = 0.1$ (dotted lines) and $\alpha = 0.01$ (dot-dashed lines).
In figure 3.1(a) we show the matrix element squared and the subtraction term as a function of $1 - x_{\bar{q}}$, where $x_{\bar{q}}$ is the fraction of the energy carried by the anti-quark, $x_{\bar{q}} = \frac{s_{34}+s_{45}}{s_{12}} = 1 - \frac{s_{35}}{s_{12}}$, with $s_{ij} = p_i.p_j$. For this process, $e^+(p_1)e^-(p_2) \to Z \to q(p_3)\bar{q}(p_4)g(p_5)$, there are only final-final state dipoles contributing to the subtraction term. The center of mass energy is set equal to the Z boson mass $\sqrt{s} = m_Z$. To restrict the discussion to the collinear divergence only, points close to the soft divergence ($x_q = x_{\bar{q}} = 1$) have been removed by forcing $x_q + x_{\bar{q}} < 1.5$ in the generation of the phase space points.
From the upper plot it is clear that both the matrix element squared and the subtraction term diverge in the collinear limit $x_{\bar{q}} \to 1$, as $1/x_{\bar{q}}$. The ratio $D/|M_R|^2$ goes to 1 and the average values of the differences fluctuate close to 0 as can be seen in the second and third plots from the top. The numerical fluctuations for small $1 - x_{\bar{q}}$ can be completely explained by statistical fluctuations. They are of the order of 1% of the maximal difference given in the lower plot. As can be expected, the cancellations are not exact, which is shown by the lower plot. The maximal difference between $|M_R|^2$ and D rises like $1/\sqrt{1 - x_{\bar{q}}}$, which does not lead to a divergent phase space integral, because the integration measure is proportional to $x_{\bar{q}}$. The small peaks/fluctuations in the region for small $x_{\bar{q}}$ are due to the fact that we are approaching the other collinear limit, *i.e.*, for which the gluon is collinear to the anti-quark $x_q \to 1$, where the matrix elements squared and the subtraction term also diverge.
In figure 3.1(b) the same matrix elements and subtraction terms are presented, but as a function of the fraction of the energy carried away by the gluon $x_g = 2 - x_q - x_{\bar{q}}$.

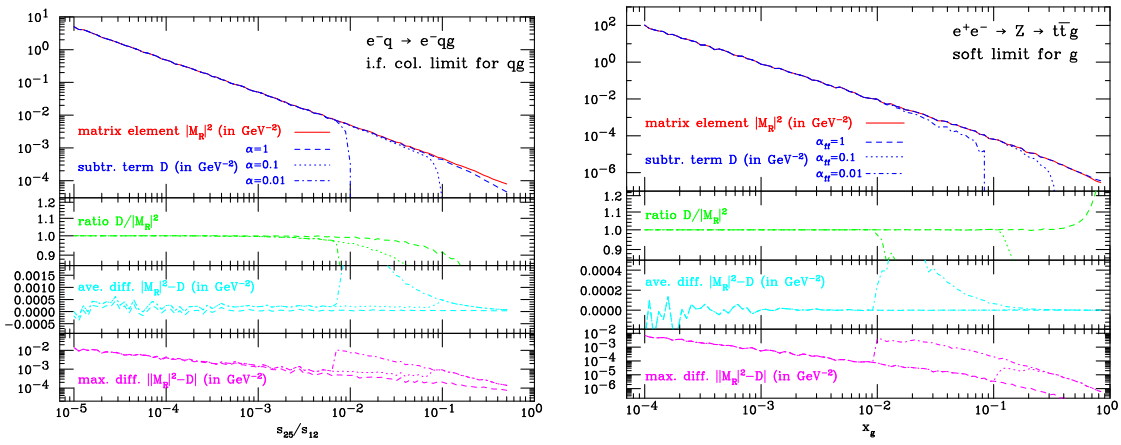


(a) (b)

Figure 3.1: Matrix element squared $|M_R|^2$ (upper plots, solid line) and the subtraction terms D (upper plots, dashed/dotted/dot-dashed lines) for the process $e^+(p_1)e^-(p_2) \to Z \to q(p_3)\bar{q}(p_4)g(p_5)$ as a function of $1 - x_{\bar{q}} = (p_3.p_5)/(p_1.p_2)$ and $x_g = 1-(p_3.p_4)/(p_1.p_2)$ in figures (a) and (b), respectively. Also plotted are the ratio $D/|M_R|^2$, the difference $|M_R|^2 - D$ (averaged over 100 random points per bin) and the maximal difference $\max(||M_R|^2 - D|)$ per bin. The dashed lines include the dipoles for each point in phase space, $\alpha_{ff} = 1$, while for the dotted $\alpha_{ff} = 0.1$ and dot-dashed $\alpha_{ff} = 0.01$ the phase space for the dipoles has been restricted to the collinear/soft regions.

The limit for which x_g goes to zero represents the soft divergence of this process, while the collinear divergences for this process are removed by excluding phase space points for which $x_{\bar{q}} > (1 + x_q)/2$ or $x_q > (1 + x_{\bar{q}})/2$. The same conclusion as for figure 3.1(a) can be drawn here: the matrix element squared and the subtraction term diverge in the soft limit, their ratio goes to one and the average difference to zero, while the absolute value of the maximal difference still rises when approaching the soft limit, but does not lead to a divergent phase space integral.

An example for a collinear limit between final state and initial state particles is given in figure 3.2(a). In this plot the matrix element squared and the subtraction term for the process $e^-(p_1)q(p_2) \to e^-(p_3)q(p_4)g(p_5)$ are given as a function of the invariant mass of the initial state quark and the final state gluon s_{25}/s_{12}. As this invariant mass goes to zero, the matrix element squared and the subtraction term diverge like $1/s_{25}$, and their ratio goes to one. To remove the other possible divergences a cut on the momentum transferred $s_{13}/s_{12} > 0.5$ and on the invariant mass of the final state quark and gluon $s_{45}/s_{12} > 0.2$ have been imposed. Like for the final-final state dipoles the average difference goes to a constant, as can be seen from the second plot from the bottom, but the dipoles have a sizeable constant contribution. Therefore the normalization of the average value for the difference $|M_R|^2 - D$ depends on the number of dipoles included for the phase space point. If all the



(a) (b)

Figure 3.2: Matrix element squared $|M_R|^2$ (upper plots, solid line) and the subtraction term D (upper plots, dashed/dotted/dot-dashed lines) for (a) the process $e^-(p_1)q(p_2) \to e^-(p_3)q(p_4)g(p_5)$ as a function of $s_{25}/s_{12} = (p_2.p_5)/(p_1.p_2)$ and (b) the process $e^+(p_1)e^-(p_2) \to Z \to t(p_3)\bar{t}(p_4)g(p_5)$ as a function of $x_g = 1-(p_3.p_4)/(p_1.p_2)$. Also plotted are the ratio $D/|M_R|^2$, the difference $|M_R|^2 - D$ (averaged over 100 random points per bin) and the maximal difference $\max(||M_R|^2 - D|)$ per bin. The dashed lines include the dipoles for each point in phase space, $\alpha = 1$, while for the dotted $\alpha = 0.1$ and dot-dashed $\alpha = 0.01$ the phase space for the dipoles has been restricted to the collinear/soft regions.

dipoles are included for all points the difference goes to a smaller constant than if we restrict the phase space of the subtraction term to be close to the singularities by setting $\alpha < 1$. Due to this restriction only the dipoles to cancel that divergence are included in the subtraction term and therefore give a smaller constant contribution, hence the difference $|M_R|^2 - D$ is larger. Also here the maximal left-over difference, the lowest plot, increases for small invariant masses but does not lead to a divergent phase space integral.
In figure 3.2(b) an example with massive final state particles is shown. The process is $t\bar{t}$ production at a linear collider, $e^+(p_1)e^-(p_2) \to t(p_3)\bar{t}(p_4)g(p_5)$ at 1 TeV center of mass energy. The plot shows a behavior very similar to the massless case, fig. 3.1, and the conclusions drawn there apply to this plot as well.

As a further check we have tested the code extensively against MCFM [25, 26]. We have generated random points in phase space and compared the subtraction terms calculated by MCFM with the subtraction terms calculated by our code. See table 3.1 for a list of processes that have been checked. We observed differences only in the case where dipoles were introduced entirely to cancel collinear limits, which can be made independently of the spectator particle. In our code all possible dipoles are calculated, which implies a sum over all spectator particles. However, if there is only a collinear divergence, *i.e.*, the unresolved parton cannot go soft, this sum is

process	subprocesses
Drell-Yan (W)	$q\bar{q}' \to W^+(\to e^+\nu_e)g$ $qg \to W^+(\to e^+\nu_e)q'$
Drell-Yan (Z)	$q\bar{q} \to Z(\to e^+e^-)g$ $qg \to Z(\to e^+e^-)q$
Drell-Yan (Z+jet)	$q\bar{q} \to Z(\to e^+e^-)q'\bar{q}'$ $q\bar{q} \to Z(\to e^+e^-)q\bar{q}$ $q\bar{q} \to Z(\to e^+e^-)gg$ $q\bar{g} \to Z(\to e^+e^-)qg$ $g\bar{g} \to Z(\to e^+e^-)q\bar{q}$
top quark pair ($t\bar{t}$)	$q\bar{q} \to t(\to bl^+\nu_l)\bar{t}(\to \bar{b}l^-\bar{\nu}_l)g$ $qg \to t(\to bl^+\nu_l)\bar{t}(\to \bar{b}l^-\bar{\nu}_l)q$ $gg \to t(\to bl^+\nu_l)\bar{t}(\to \bar{b}l^-\bar{\nu}_l)g$
t-channel single top with massive b-quark [96]	$gg \to t\bar{b}q\bar{q}'$ $qq' \to t\bar{b}q'q''$ $q\bar{q}' \to t\bar{b}q'q''$ $qg \to t\bar{b}q'g$

Table 3.1: Set of processes for which the MadDipole code has been tested against MCFM for random points in phase space. All the possible initial-initial, initial-final and final-initial, dipoles for massless and massive final state particles have been checked with this set of subprocesses. No inconsistencies were found.

redundant and one dipole with the appropriate coefficient is enough to cancel the singularity. In MCFM, these special limits are implemented using a single spectator momentum, while MadDipole sums over all spectator momenta, thereby yielding a different subtraction term. We have checked in the relevant cases that close to the singularities the MCFM subtraction terms behave identical to the subtraction terms calculated by our code. Furthermore our code has been tested against the HELAC implementation [91] and with [62]
We also tested the CPU time which is needed to produce the squared matrix element and the dipoles for a given phase space point. These checks were performed with an Intel Pentium 4 processor with 3.20GHz. As an example we picked out three different processes:

1) $gg \to gggg$: $|\mathcal{M}|^2$: 26ms, $\sum_{dipoles}$: 68ms

2) $u\bar{u} \to d\bar{d}ggg$: $|\mathcal{M}|^2$: 10ms, $\sum_{dipoles}$: 45ms

3) $u\bar{u} \to u\bar{u}ggg$: $|\mathcal{M}|^2$: 34ms, $\sum_{dipoles}$: 0.15s

The time which is needed to produce the `Fortran` code is strongly dependent on the process and ranges from a few seconds to at most a few minutes. The process $gg \to 5g$ is currently not yet feasible within MadGraph because of the size of the

color factors. Once MadGraph has been adjusted to handle this process, it will equally become accessible for MadDipole.

Chapter 4

Integrated Dipoles

4.1 Integration over one particle phase space

In the previous chapter we explained the implementation of an additional parameter α with $\alpha \in [0, 1]$ introduced in [92] and [27]. This parameter cuts away parts of the phase space where there is no singularity and therefore no subtraction terms are needed. But this parameter has an impact on the integrated dipoles which lead to a dependence on α of the integrated terms. Most of the results already existed in the literature [27, 92–94] and we confirm them, for the other cases new results have been derived.
In the following Q denotes a massive quark whereas q is a massless one. Furthermore we always assume that the process contains only one mass scale. That means if emitter and spectator are massive particles their masses are assumed to be equal.
The introduction of an α-parameter leads to a reduction of the upper limit of the corresponding integration variable. However the lower limit is not affected. We calculate the corrections coming from the α cut in such a way that we take the full result which is known and then subtract the integral between the new upper limit and the original one. The advantage of this method is that we can perform the limit $\epsilon \to 0$ before the integration which leads to a finite result for our correction terms. This is possible because divergencies only occur at the lower integration limit but this limit does not affect our integration.

4.1.1 Final-final

The one particle phase space where both emitter and spectator are final state particles is given in Eq.(5.20) in Ref. [12] for massless particles and in Eq.(5.11) in Ref. [13] for massive partons.
We now have to distinguish between several possibilities:

a) $Q \to Q + g$ with massive spectator,

b) $Q \to Q + g$ with massless spectator,

c) $q \to q \ + \ g$ with massive spectator,

d) $q \to q \ + \ g$ with massless spectator,

e) $g \to g \ + \ g$ with massive spectator,

f) $g \to g \ + \ g$ with massless spectator,

g) $g \to Q \ + \ \bar{Q}$ with massive spectator,

h) $g \to Q \ + \ \bar{Q}$ with massless spectator,

i) $g \to q \ + \ \bar{q}$ with massive spectator,

j) $g \to q \ + \ \bar{q}$ with massless spectator.

Phase space restrictions for final-final dipoles

The cases (b) and (c) including the α-parameter can be found in Ref. [93], cases (d), (f) and (j) are calculated in [92]. We are now going to discuss the remaining cases (a),(e),(g),(h) and (i).
We start with case (a), the splitting $Q \to Q \ + g$ with a massive spectator. The splitting function is given by

$$\begin{aligned}\langle s|\mathbf{V}_{gQ,k}|s'\rangle \quad &= 8\pi\mu^{2\epsilon}\alpha_{\rm s}C_{\rm F}\left\{\frac{2}{1-\tilde{z}_j(1-y_{ij,k})} - \frac{\tilde{v}_{ij,k}}{v_{ij,k}}\left[1+\tilde{z}_j+\frac{m_Q^2}{p_ip_j}+\epsilon(1-\tilde{z}_j)\right]\right\}\delta_{ss'} \\ &= \langle\mathbf{V}_{gQ,k}\rangle\delta_{ss'}\ , \end{aligned} \tag{4.1}$$

where $\langle\mathbf{V}_{gQ,k}\rangle$ is the spin averaged splitting function. The integral over the one particle phase space can be defined as (Eq.(5.22) CDST)

$$\int[{\rm d}p_i(\widetilde{p}_{ij},\widetilde{p}_k)]\,\frac{1}{(p_i+p_j)^2-m_{ij}^2}\,\langle\mathbf{V}_{ij,k}\rangle \ \equiv\ \frac{\alpha_{\rm s}}{2\pi}\frac{1}{\Gamma(1-\epsilon)}\left(\frac{4\pi\mu^2}{Q^2}\right)^{\epsilon} I_{ij,k}(\epsilon)\ , \tag{4.2}$$

where the phase space is given by the following expression (Eq.(5.11), CDST):

$$\begin{aligned}\int[{\rm d}p_i(\widetilde{p}_{ij},\widetilde{p}_k)] \quad &= \frac{1}{4}(2\pi)^{-3+2\epsilon}(Q^2)^{1-\epsilon}(1-\mu_i^2-\mu_j^2-\mu_k^2)^{2-2\epsilon}\left[\lambda(1,\mu_{ij}^2,\mu_k^2)\right]^{\frac{-1+2\epsilon}{2}}\int{\rm d}^{d-3}\Omega \\ &\times\int_{y_-}^{y_+}{\rm d}y_{ij,k}\,(1-y_{ij,k})^{1-2\epsilon}\left[\mu_i^2+\mu_j^2+(1-\mu_i^2-\mu_j^2-\mu_k^2)y_{ij,k}\right]^{-\epsilon} \\ &\times\int_{z_-(y_{ij,k})}^{z_+(y_{ij,k})}{\rm d}\tilde{z}_i\ \left[z_+(y_{ij,k})-\tilde{z}_i\right]^{-\epsilon}\left[\tilde{z}_i-z_-(y_{ij,k})\right]^{-\epsilon}\ . \end{aligned} \tag{4.3}$$

Without the cut parameter, the integration ranges are given by

$$\begin{aligned} 0 \;&<\; y_{ij,k} \;< \frac{1-2\mu_Q}{1-2\mu_Q^2} \\ \frac{\left(2\mu_Q^2-1\right)(v_{ij,k}-1)y_{ij,k}}{\mu_Q^2(2-4y_{ij,k})+2y_{ij,k}} \;&<\; \tilde{z}_i \;< \frac{\left(2\mu_Q^2-1\right)(v_{ij,k}+1)y_{ij,k}}{\mu_Q^2(4y_{ij,k}-2)-2y_{ij,k}}. \end{aligned} \tag{4.4}$$

We denote the upper/lower limits of the integration variables with $y_{+/-}$ and $z_{+/-}$ respectively. The introduction of the α-parameter now imposes an additional Θ-function of the form $\Theta(\alpha y_+ - y_{ij,k})$ the phase space in (4.3) is multiplied with. As we are only interested in the correction pieces, our integration range for the y integration is given by

$$\alpha y_+ \;<\; y_{ij,k} \;<\; y_+. \tag{4.5}$$

The integral over y can be performed with the help of the transformation

$$y' = \left(1-2\mu_Q^2\right)y - \sqrt{4\mu_Q^4y^2 - 4\mu_Q^2\left(y^2-y+1\right)+(y-1)^2}+1. \tag{4.6}$$

Performing the integration we find

$$\begin{aligned} I_{ij,k}(\epsilon,\alpha) = I_{ij,k}(\epsilon) - C_{\mathrm{F}}\Bigg(&\left((\alpha-1)\left(3\alpha(\mu_Q-1)-2\mu_Q^2\right)(1-2\mu_Q)^2+(\mu_Q-1)\left(\mu_Q^2-2\alpha\mu_Q+\alpha\right)\right. \\ &\left.\left(\left(8\mu_Q^2-4\right)\log(\alpha)+\log\left(\frac{\mu_Q^2-2\alpha\mu_Q+\alpha}{(\mu_Q-1)^2}\right)\right)\right)\left(2(\mu_Q-1)\left(\mu_Q^2-2\alpha\mu_Q+\alpha\right)\right. \\ &\left.\left(2\mu_Q^2-1\right)\right)+\Bigg(\left(2\mu_Q^2-1\right)\Bigg(\log^2(2\mu_Q)-2\log\left(-8\mu_Q(2\mu_Q-1)\left(\sqrt{1-4\mu_Q^2}+1\right)\right) \\ &\log(2\mu_Q)-\log^2(2-y'_-)-2\log(2-2\mu_Q)\log\left(-4\mu_Q(2\mu_Q-1)\left(\sqrt{1-4\mu_Q^2}+1\right)\right) \\ &-2\log\left(\frac{\mu_Q-1}{\sqrt{1-4\mu_Q^2}-1}\right)\log\left(-2\mu_Q+\sqrt{1-4\mu_Q^2}+1\right)+2\log\left(-\frac{(\mu_Q-1)\left(\sqrt{1-4\mu_Q^2}+1\right)}{\mu_Q^2}\right) \\ &\log\left(2\mu_Q+\sqrt{1-4\mu_Q^2}-1\right)+2\log(-2(\mu_Q-1)\mu_Q)\log\left((4-8\mu_Q)\mu_Q\left(-2\mu_Q^2+\sqrt{1-4\mu_Q^2}+1\right)\right) \\ &-2\log\left(-y'_-+\sqrt{1-4\mu_Q^2}+1\right)\log\left(\frac{\left(\sqrt{1-4\mu_Q^2}+1\right)y'_-}{2\mu_Q^2}\right) \\ &-\log(\mu_Q)\left(4\log\left(\frac{2(\mu_Q-1)\mu_Q}{2\mu_Q^2+y'_--2}\right)+\log(4)\right)-2\log\left(-\frac{\left(\sqrt{1-4\mu_Q^2}-1\right)(y'_--2)}{2\mu_Q^2+y'_--2}\right) \\ &\log\left(y'_-+\sqrt{1-4\mu_Q^2}-1\right)+2\log\left(\frac{2\mu_Q^2+y'_--2}{(y'_--2)y'_-}\right)\log\left(\frac{4\mu_Q^2}{y'_--2}+y'_-\right) \\ &+\log(4)\log\left(\frac{8\mu_Q^3}{y'_--2}+2y'_-\mu_Q\right)+2\log(-(y'_--2)y'_-)\log\left(-\left(\sqrt{1-4\mu_Q^2}+1\right)\left(4\mu_Q^2+(y'_--2)y'_-\right)\right) \\ &-2\log\left(-2\mu_Q^2-y'_-+2\right)\log\left(-\left(-2\mu_Q^2+\sqrt{1-4\mu_Q^2}+1\right)\left(4\mu_Q^2+(y'_--2)y'_-\right)\right) \end{aligned}$$

$$
\begin{aligned}
&-2\mathrm{Li}_2(1-\mu_Q)+2\mathrm{Li}_2\left(\frac{\mu_Q-1}{\mu_Q}\right)+2\mathrm{Li}_2\left(\frac{2-2\mu_Q}{\sqrt{1-4\mu_Q^2}+1}\right)+2\mathrm{Li}_2\left(\frac{2\mu_Q}{\sqrt{1-4\mu_Q^2}+1}\right)\\
&-2\mathrm{Li}_2\left(-\frac{-2\mu_Q+\sqrt{1-4\mu_Q^2}+1}{2\mu_Q}\right)-2\mathrm{Li}_2\left(-\frac{2(\mu_Q-1)\mu_Q}{-2\mu_Q^2+\sqrt{1-4\mu_Q^2}+1}\right)\\
&+2\mathrm{Li}_2\left(\frac{2\mu_Q+\sqrt{1-4\mu_Q^2}-1}{2\mu_Q^2+\sqrt{1-4\mu_Q^2}-1}\right)-2\mathrm{Li}_2\left(-\frac{-4\mu_Q^2-2\mu_Q-2(\mu_Q-1)\sqrt{1-4\mu_Q^2}+2}{4\mu_Q^2}\right)\\
&-2\mathrm{Li}_2\left(\frac{2-y'_-}{\sqrt{1-4\mu_Q^2}+1}\right)+2\mathrm{Li}_2\left(\frac{-y'_-+\sqrt{1-4\mu_Q^2}+1}{\sqrt{1-4\mu_Q^2}-1}\right)-2\mathrm{Li}_2\left(\frac{-y'_-+\sqrt{1-4\mu_Q^2}+1}{2\mu_Q^2+\sqrt{1-4\mu_Q^2}-1}\right)\\
&+2\mathrm{Li}_2\left(\frac{y'_-}{2}\right)-2\mathrm{Li}_2\left(\frac{y'_-}{\sqrt{1-4\mu_Q^2}+1}\right)-2\mathrm{Li}_2\left(\frac{2\mu_Q^2+y'_--2}{2\mu_Q^2}\right)\\
&+2\mathrm{Li}_2\left(-\frac{2\mu_Q^2+y'_--2}{-2\mu_Q^2+\sqrt{1-4\mu_Q^2}+1}\right)+2\mathrm{Li}_2\left(-\frac{-4\mu_Q^2+\sqrt{1-4\mu_Q^2}y'_-+y'_-}{4\mu_Q^2}\right)\\
&+\tanh^{-1}\left(\frac{y'_-}{\mu_Q^2-1}+1\right)\log(16)\Bigg)\Bigg)\Big/\left(\sqrt{1-4\mu_Q^2}\right)\Bigg)\ ,
\end{aligned}
\tag{4.7}
$$

where y'_- is given by

$$
y'_- = -2\alpha\mu_Q-\sqrt{(\alpha-1)(2\mu_Q-1)(2(\alpha+1)\mu_Q-\alpha+1)}+\alpha+1. \tag{4.8}
$$

With the same procedure we confirm the results for cases (b) and (c) found in Ref. [93].

For case (e) the spin averaged splitting function for $g\to g\,g$ and a massive spectator is given by (Eq.5.20 CDST)

$$
\begin{aligned}
\langle \mathbf{V}_{gg,k}\rangle \quad &= 16\pi\mu^{2\epsilon}\alpha_{\mathrm{s}}C_{\mathrm{A}}\left\{\frac{1}{1-\tilde{z}_i(1-y_{ij,k})}+\frac{1}{1-\tilde{z}_j(1-y_{ij,k})}\right.\\
&\qquad\left.+\frac{\tilde{z}_i(1-\tilde{z}_i)-(1-\kappa)z_+z_--2}{v_{ij,k}}\right\}.
\end{aligned}
\tag{4.9}
$$

Four our calculation we set κ to zero. The integration boundaries are

$$
\begin{aligned}
0 \ &< y_{ij,k} \ < \frac{1-\mu_k}{\mu_k+1}\\
\frac{1-v_{i,jk}}{2} \ &< \tilde{z}_i \ < \frac{1+v_{i,jk}}{2},
\end{aligned}
\tag{4.10}
$$

where again with $y_{+/-}$ and $z_{+/-}$ we denote the integration limits given in the inequalities above. Setting the additional constraint $\Theta(\alpha y_+ - y_{ij,k})$ in the phase space

of Eq.4.3 and using

$$y' = \mu_k^2(-y-1) - \sqrt{(\mu_k^2-1)\left(\mu_k^2(y+1)^2-(y-1)^2\right)} + y + 1 \tag{4.11}$$

yields to the following result:

$$\begin{aligned}
I_{ij,k}(\epsilon,\alpha) = I_{ij,k}(\epsilon) - C_{\mathrm{A}} \Bigg(& \frac{1}{6}\Bigg(48\mathrm{Li}_2\left(\frac{y'_-}{2-2\mu_k^2}\right) + \frac{11(2\mu_k+y'_--2)^2}{(\mu_k^2-1)(y'_--2)} \\
& -12\Bigg(-\log^2\left(-\frac{y'_-}{\mu_k^2-1}\right) + \log^2\left(-\frac{2\left(2\mu_k^2+y'_--2\right)}{(\mu_k^2-1)(y'_--2)}\right) + 4\log\left(1-\frac{y'_-}{2}\right)\log\left(-\frac{y'_-}{\mu_k^2-1}\right) \\
& +2\left(\log\left(-\frac{2\left(2\mu_k^2+y'_--2\right)}{(\mu_k^2-1)(y'_--2)}\right) - 2\log\left(\frac{y'_-}{2\mu_k^2-2}+1\right)\right)\log\left(-\frac{y'_-}{\mu_k^2-1}\right)\Bigg) \\
& + \Big(2\left(4\mu_k^2\left(\log\left(4\mu_k^2(y'_--1)+(y'_--2)^2\right) - \log\left(-8(\mu_k-1)\mu_k^2\right)\right) + \left(11-15\mu_k^2\right)\right. \\
& \left.\log(2-y'_-) + \left(15\mu_k^2-11\right)\log(2\mu_k)\right)\Big) / \left(\mu_k^2-1\right) + 22\log\left(-2\mu_k^2-y'_-+2\right) + 48\mathrm{Li}_2(1-\mu_k) \\
& -48\mathrm{Li}_2\left(\frac{1}{\mu_k+1}\right) - 44\log(2-2\mu_k) - 22\log(\mu_k) + 24\log\left(\frac{2}{\mu_k+1}\right) \\
& \left(\log\left(\frac{2}{\mu_k+1}\right) + 2\log(\mu_k+1)\right) - 48\mathrm{Li}_2\left(\frac{y'_-}{2}\right) + 22\log(y'_-)\Bigg)\Bigg),
\end{aligned} \tag{4.12}$$

with

$$y'_- = -\sqrt{(\mu_k-1)^2\left(\alpha^2(\mu_k-1)^2 - 2\alpha\left(\mu_k^2+1\right) + (\mu_k+1)^2\right)} + \alpha(\mu_k-1)^2 - \mu_k^2 + 1. \tag{4.13}$$

Case (g) is the splitting of a gluon into two massive quarks ($g \to Q\ \bar{Q}$) with a massive spectator. The splitting function for that process is (Eq.5.18 CDST)

$$\begin{aligned}
\langle \mathbf{V}_{Q\bar{Q},k} \rangle = 8\pi\mu^{2\epsilon}\alpha_{\mathrm{s}}T_{\mathrm{R}}\frac{1}{v_{ij,k}}\Bigg\{ 1 - \frac{2}{1-\epsilon}\Bigg[& \tilde{z}_i(1-\tilde{z}_i) - (1-\kappa)z_+z_- \\
& - \frac{\kappa\mu_Q^2}{2\mu_Q^2 + (1-2\mu_Q^2-\mu_k^2)y_{ij,k}}\Bigg]\Bigg\},
\end{aligned} \tag{4.14}$$

where we neglect the last term because we set $\kappa = 0$. As already written above for this case we assume that mass of the spectator is the same as the mass occurring in the splitting. Under this assumption the integration ranges are

$$\frac{2\mu_k^2}{1-3\mu_k^2} \ < y_{ij,k} \ < \ 1 - \frac{2\mu_k(1-\mu_k)}{1-3\mu_k^2}, \tag{4.15}$$

$$\frac{1-v_{ij,i}v_{ij,k}}{2} \ < \tilde{z}_i \ < \frac{1+v_{ij,i}v_{ij,k}}{2}. \tag{4.16}$$

Here is an important difference to the two cases discussed so far. In this case the integration over $y_{ij,k}$ does not start at $y_{ij,k} = 0$ but at a value larger than zero. But introducing the alpha parameter and calculating the correction terms means new integration boundaries for the $y_{ij,k}$ integration according to Eq.(4.5). The result of the integration is only a sensible value if

$$\frac{2\mu_k^2}{1-3\mu_k^2} \;<\; \alpha y_+. \tag{4.17}$$

That means that the value for α must not be chosen to be too small. Keeping this in mind we find

$$\begin{aligned}
I_{ij,k}(\epsilon,\alpha) = I_{ij,k}(\epsilon) - 2T_{\mathrm{R}} \Bigg(\Bigg(\Bigg(2 \Bigg(& \sqrt{\alpha^2(\mu_k(\mu_k+2)-1)^2 - 4\mu_k^4} + \mu_k^2 \Bigg(-5\sqrt{\alpha^2(\mu_k(\mu_k+2)-1)^2 - 4\mu_k^4} \\
& + \mu_k \Bigg(\mu_k \Bigg(6\sqrt{\alpha^2(\mu_k(\mu_k+2)-1)^2 - 4\mu_k^4} + 7\pi\mu_k^2 + 9\sqrt{(1-3\mu_k)(\mu_k+1)}\mu_k \\
& + 3\sqrt{(1-3\mu_k)(\mu_k+1)} - 6\pi \Bigg) - 3\sqrt{(1-3\mu_k)(\mu_k+1)} \Bigg) - \sqrt{(1-3\mu_k)(\mu_k+1)} + \pi \Bigg) \Bigg) \\
& - \alpha(\mu_k(\mu_k+2)-1) \Bigg(\mu_k \Bigg(\mu_k \Bigg(3 \Bigg(\sqrt{\alpha^2(\mu_k(\mu_k+2)-1)^2 - 4\mu_k^4} + \sqrt{(1-3\mu_k)(\mu_k+1)} \\
& - 2\pi \Bigg) + \mu_k \Bigg(7\pi\mu_k + 9\sqrt{(1-3\mu_k)(\mu_k+1)} \Bigg) \Bigg) - 3\sqrt{(1-3\mu_k)(\mu_k+1)} \Bigg) \\
& - \sqrt{\alpha^2(\mu_k(\mu_k+2)-1)^2 - 4\mu_k^4} - \sqrt{(1-3\mu_k)(\mu_k+1)} + \pi \Bigg) \Bigg) \Big/ \left(\alpha(\mu_k(\mu_k+2)-1) - 2\mu_k^2\right) \\
& + 2\left(7\mu_k^4 - 6\mu_k^2 + 1\right) \sec^{-1}\left(\frac{\alpha(\mu_k(\mu_k+2)-1)}{\sqrt{\alpha^2(\mu_k(\mu_k+2)-1)^2 - 4\mu_k^4}} \right) + \left(2\left(-7\mu_k^6 + 13\mu_k^4 - 7\mu_k^2 + 1\right)\right. \\
& \left. \sin^{-1}\left(\frac{(\mu_k-1)\sqrt{(1-3\mu_k)(\mu_k+1)}}{\mu_k(\mu_k+2)-1} \right) - \left(5\mu_k^2 - 1\right) \left(2\sqrt{5\mu_k^4 - 6\mu_k^2 + 1}\mu_k^2 \right.\right. \\
& \log\Bigg((2-2\mu_k)\mu_k \Bigg(\sqrt{\left(5\mu_k^4 - 6\mu_k^2 + 1\right)\left(\alpha^2(\mu_k(\mu_k+2)-1)^2 - 4\mu_k^4\right)} \\
& + \alpha\left(3\mu_k^2 - 1\right)\left(\mu_k(\mu_k+2)-1\right) - 4\mu_k^4 \Bigg) \Bigg) + \left(-3\mu_k^4 + 4\mu_k^2 - 1\right) \\
& \log\Bigg(-2\left(\sqrt{\alpha^2(\mu_k(\mu_k+2)-1)^2 - 4\mu_k^4} + \alpha(\mu_k(\mu_k+2)-1) \right) \Bigg) - 2\sqrt{5\mu_k^4 - 6\mu_k^2 + 1}\mu_k^2 \\
& \log\Bigg(\Bigg(-\mu_k^4 + 6\mu_k^3 - 4\mu_k^2 + (\mu_k+1)\sqrt{-(\mu_k-1)^3(3\mu_k-1)\left(5\mu_k^2-1\right)} - 2\mu_k + 1 \Bigg) \\
& \left(\alpha(\mu_k(\mu_k+2)-1) - 3\mu_k^2 + 1\right) \Big) + \left(3\mu_k^4 - 4\mu_k^2 + 1\right) \\
& \log\left(-2\left(\mu_k^2 + 2\mu_k - (\mu_k-1)\sqrt{(1-3\mu_k)(\mu_k+1)} - 1 \right) \right) \Bigg) \Bigg) \Big/ \left(\mu_k^2 - 1\right) \Bigg) \\
& \Big/ \left(3\left(3\mu_k^4 - 4\mu_k^2 + 1\right)\right) \Bigg).
\end{aligned} \tag{4.18}$$

Case (h) considers also the splitting of a gluon into a massive quark pair ($g \to Q\,\bar{Q}$) however with a massless spectator. While the splitting function is the same

(Eq.4.14), the integration boundaries are different, namely

$$\frac{2\mu_i^2}{1-2\mu_i^2} \; < y_{ij,k} \; < 1 \tag{4.19}$$

$$\frac{1-v_{ij,i}}{2} \; < \tilde{z}_i \; < \; \frac{1+v_{ij,i}}{2}. \tag{4.20}$$

The phase space integral gets multiplied by an additional factor of $\Theta(\alpha - y_{ij,k})$ and integration leads to

$$\begin{aligned} I_{ij,k}(\epsilon,\alpha) = I_{ij,k}(\epsilon) - T_{\mathrm{R}} \Bigg(\frac{2}{3} \Bigg(& \frac{2\sqrt{\alpha^2\left(1-2\mu_j^2\right)^2 - 4\mu_j^4}}{2(\alpha-1)\mu_j^2 - \alpha} + \sqrt{\alpha^2\left(1-2\mu_j^2\right)^2 - 4\mu_j^4} + \left(2\mu_j^2 - 1\right) \\ & \left(-\log\left(-2\left(\sqrt{\alpha^2\left(1-2\mu_j^2\right)^2 - 4\mu_j^4} + \alpha\left(2\mu_j^2-1\right)\right)\right) + 2\tan^{-1}\left(\frac{2\mu_j^2}{\sqrt{\alpha^2\left(1-2\mu_j^2\right)^2 - 4\mu_j^4}}\right) \right. \\ & \left. + \log\left(-2\left(2\mu_j^2 + \sqrt{1-4\mu_j^2} - 1\right)\right) - 2\tan^{-1}\left(\frac{2\mu_j^2}{\sqrt{1-4\mu_j^2}}\right)\right) + \sqrt{1-4\mu_j^2}\Bigg)\Bigg). \end{aligned} \tag{4.21}$$

Finally we come to the splitting of a gluon into two massless quarks ($g \to q\ \bar{q}$) with a massive spectator. The integration ranges for this possibility are

$$0 \; < y_{ij,k} \; < \; 1 - \frac{2\mu_l(1-\mu_k)}{1-\mu_k^2} \tag{4.22}$$

$$\frac{1-v_{ij,k}}{2} \; < \tilde{z}_i \; < \; \frac{1+v_{ij,k}}{2}. \tag{4.23}$$

Performing the integration leads to

$$\begin{aligned} I_{ij,k}(\epsilon,\alpha) = I_{ij,k}(\epsilon) - T_{\mathrm{R}} \Bigg(\Bigg(2\Bigg(\mu_k^2 \Bigg(& -2\log\left(\alpha\left(\frac{2}{\mu_k+1} - 1\right) - 1\right) + \log\left(\frac{4}{\alpha}\right) \\ & + 2\log\left(\frac{1}{\mu_k+1} - 1\right)\Bigg) - (\alpha-1)(\mu_k-1)^2 + \log(\alpha)\Bigg)\Bigg) / \left(3\left(\mu_k^2-1\right)\right)\Bigg). \end{aligned} \tag{4.24}$$

4.1.2 Final-initial

Now we consider the case where the emitter is a final state particle and the spectator is in the initial state. We assume that initial state particles are always massless. So we have the following possibilities:

a) $\tilde{ij}: \; Q \qquad$ emitter : $\; Q$,

b) $\tilde{ij}: \; q \qquad$ emitter : $\; q$,

c) $\tilde{ij}: g \qquad \text{emitter}: Q,$

d) $\tilde{ij}: g \qquad \text{emitter}: q,$

e) $\tilde{ij}: g \qquad \text{emitter}: g,$

The results for (a) and (b) are given in Ref. [93] so only the three remaining cases (c-e) need to be discussed.

Phase space restrictions for final-initial dipoles

The one particle phase space for final-initial dipoles is given in Eq.(5.48) CDST:

$$\int [\mathrm{d}p_i(\widetilde{p}_{ij}; p_a, x)] = \frac{1}{4}(2\pi)^{-3+2\epsilon}(2\widetilde{p}_{ij}p_a)^{1-\epsilon} \int_0^{x_+} \mathrm{d}x_{ij,a}\, \delta(x - x_{ij,a})\, (1 - x + \mu_{ij}^2)^{-\epsilon} \times \int \mathrm{d}^{d-3}\Omega \int_{z_-(x)}^{z_+(x)} \mathrm{d}\tilde{z}_i\, [z_+(x) - \tilde{z}_i]^{-\epsilon}\, [\tilde{z}_i - z_-(x)]^{-\epsilon}\,, \tag{4.25}$$

and the integrated splitting function is defined via Eq.(5.53) CDST:

$$\int [\mathrm{d}p_i(\widetilde{p}_{ij}; p_a, x)]\, \frac{1}{(p_i + p_j)^2 - m_{ij}^2}\, \langle \mathbf{V}_{ij}^a \rangle \equiv \frac{\alpha_\mathrm{s}}{2\pi} \frac{1}{\Gamma(1-\epsilon)} \left(\frac{4\pi\mu^2}{2\widetilde{p}_{ij}p_a} \right)^{\epsilon} I_{ij}^a(x;\epsilon)\,. \tag{4.26}$$

For case (c), the splitting of a gluon into massive quarks ($g \to Q\,\bar{Q}$), the integrated spitting function is given in Eq.(5.57) CDST:

$$I_{Q\bar{Q}}^a(x;\epsilon) = T_\mathrm{R} \left\{ [J_{Q\bar{Q}}^a(x,\mu_Q)]_{x_+} + \delta(x_+ - x) \left[J_{Q\bar{Q}}^{a;\mathrm{S}}(\mu_Q;\epsilon) + J_{Q\bar{Q}}^{a;\mathrm{NS}}(\mu_Q) \right] \right\} + \mathrm{O}(\epsilon)\,, \tag{4.27}$$

where the x_+-distribution is defined as (Eq.(5.55) CDST)

$$\int_0^1 \mathrm{d}x \left(f(x) \right)_{x_+} g(x) \equiv \int_0^1 \mathrm{d}x\, f(x) \Theta(x_+ - x)\, [g(x) - g(x_+)]\,. \tag{4.28}$$

Imposing the cut on the α-parameter implies that the phase space in (4.25) is multiplied with $\Theta(\alpha - 1 + x_{ija})$.
This leads to a modification of the x_+-distribution terms and we get in analogy to Eq.(5.62) CDST

$$[J_{Q\bar{Q}}^a(x, \mu_Q, \alpha)]_{x_+} = \frac{2}{3} \left(\frac{1 - x + 2\mu_Q^2}{(1-x)^2} \sqrt{1 - \frac{4\mu_Q^2}{1-x}} \right)_{1-\alpha}\,, \tag{4.29}$$

where we define:

$$\int_0^1 dx\; f(x)\, (g(x))_{1-\alpha} = \int_{1-\alpha}^1 dx\; g(x)\, (f(x) - f(1)) \tag{4.30}$$

For the non-singular terms $J^{a;\mathrm{NS}}_{Q\bar{Q}}(\mu_Q)$ we find the following modifications:

$$J^{a;\mathrm{NS}}_{Q\bar{Q}}(\mu_Q,\alpha) = J^{a;\mathrm{NS}}_{Q\bar{Q}}(\mu_Q) + \frac{2}{9}\left(\left(-4\mu_Q^2\left(\sqrt{\frac{\left(1-4\mu_Q^2\right)\left(\alpha-4\mu_Q^2\right)}{\alpha^3}}+4\right) - 5\sqrt{\frac{\left(1-4\mu_Q^2\right)\left(\alpha-4\mu_Q^2\right)}{\alpha}}\right.\right.$$
$$\left.\left.-16\mu_Q^4+5\right) / \left(\sqrt{1-4\mu_Q^2}\right) + 6\log\left(\sqrt{\alpha-4\mu_Q^2}+\sqrt{\alpha}\right) - 6\log\left(\sqrt{1-4\mu_Q^2}+1\right)\right). \tag{4.31}$$

Case (d) is just the limit $\mu_Q \to 0$ of (4.29), i.e.

$$[J^a_{q\bar{q}}(x,0,\alpha)]_+ = \frac{2}{3}\left(\frac{1}{1-x}\right)_{1-\alpha}, \tag{4.32}$$

which leads to the following additional non-singular terms:

$$J^{a;\mathrm{NS}}_{q\bar{q}}(0,\alpha) = J^{a;\mathrm{NS}}_{q\bar{q}}(0) + \frac{2}{3}\log\alpha. \tag{4.33}$$

In the case of the splitting ($g \to g\ g$) the general structure of the integrated splitting function is (Eq.(5.66) CDST)

$$I^a_{gg}(x;\epsilon) = 2C_\mathrm{A}\left\{[J^a_{gg}(x)]_+ + \delta(1-x)\,J^{a;\mathrm{S}}_{gg}(\epsilon)\right\} + \mathrm{O}(\epsilon)\,. \tag{4.34}$$

The first term $[J^a_{gg}(x)]_+$ contains all +-distributions but is not a +-distribution itself. In the presence of the α-parameter we find

$$[J^a_{gg}(x,\alpha)]_+ = \left(\frac{2}{1-x}\ln\frac{1}{1-x} - \frac{11}{6}\frac{1}{1-x}\right)_{1-\alpha} + \frac{2}{1-x}\ln(2-x)\Theta(\alpha-1+x), \tag{4.35}$$

which leads to a modification of the terms proportional to $\delta(1-x)$ of Eq.(5.68) CDST of the following form:

$$J^{a;\mathrm{S}}_{gg}(x,\alpha) = J^{a;\mathrm{S}}_{gg}(x) - \log^2\alpha - \frac{11}{6}\log\alpha. \tag{4.36}$$

4.1.3 Initial-final

Next we have the case where the emitter is an initial state particle and the spectator is in the final state. Again we assume that initial state particles are massless. We distinguish between the processes

a) $\tilde{ij}: \; q \qquad \text{emitter}: \; q,$

b) $\tilde{ij}: \; g \qquad \text{emitter}: \; q,$

c) $\tilde{ij}: \; q \qquad \text{emitter}: \; g,$

d) $\tilde{ij}: \; g \qquad \text{emitter}: \; g.$

Concerning the α dependence the results for all of the four cases can be found in Refs. [93, 94]. From the analytical point of view the limit of a vanishing spectator mass can be performed without any problems. However taking the result for a massive spectator and setting the mass to zero in the numerical implementation causes problems for the cases (b) and (d). For these two cases we calculated the limit analytically and implemented a massive and a massless version.

4.1.4 Initial-initial

Finally we have the case where both emitter and spectator are in the initial state. Now both have to be massless and we have as for initial-final the following constellations:

a) $\tilde{ij}: \; q \qquad \text{emitter}: \; q,$

b) $\tilde{ij}: \; g \qquad \text{emitter}: \; q,$

c) $\tilde{ij}: \; q \qquad \text{emitter}: \; g,$

d) $\tilde{ij}: \; g \qquad \text{emitter}: \; g.$

Also for the initial-initial dipoles, the α-dependence has already been calculated in [93, 94].

4.2 Expansion of integrated dipoles in ϵ

The formal structure of the integrated dipoles is independent of which configuration we are considering. We pick out the final-final case but the following holds for an arbitrary configuration. In the final-final case, the integrated dipole function is written as

$$\int [\mathrm{d}p_i(\widetilde{p}_{ij}, \widetilde{p}_k)] \, \frac{1}{(p_i + p_j)^2 - m_{ij}^2} \, \langle \mathbf{V}_{ij,k} \rangle \; \equiv \; \frac{\alpha_\mathrm{s}}{2\pi} \frac{1}{\Gamma(1-\epsilon)} \left(\frac{4\pi\mu^2}{2p_i \cdot p_k} \right)^\epsilon \mathcal{V}_i(\epsilon) \, . \qquad (4.37)$$

Depending on the configuration the splitting function and the propagator of the left hand side of (4.37) is different, however the structure of the result on the right hand

side is the same.
The factor $\mathcal{V}_i(\epsilon)$ is singular in the limit $\epsilon \to 0$ so what one usually does is an expansion in ϵ of the right hand side of (4.37).
The expansion in ϵ can be written symbolically as

$$\begin{aligned}\int [\mathrm{d}p_i(\widetilde{p}_{ij},\widetilde{p}_k)]\,\frac{1}{(p_i+p_j)^2-m_{ij}^2}\,\langle \mathbf{V}_{ij,k}\rangle \;&\equiv\; \frac{\alpha_\mathrm{s}}{2\pi}\frac{1}{\Gamma(1-\epsilon)}\left(\frac{4\pi\mu^2}{2p_i\cdot p_k}\right)^{\epsilon}\mathcal{V}_i(\epsilon)\\ &= \frac{y_1}{\epsilon^2}+\frac{y_1}{\epsilon}+y_3\ ,\end{aligned} \tag{4.38}$$

where the coefficients y_{1-3} are determined by the specific process.
One straightforward possibility is to expand the whole right hand side of (4.37), i.e. including the overall prefactors. However this method has a disadvantage. Expanding the prefactor gives

$$\frac{\alpha_\mathrm{s}}{2\pi}\frac{1}{\Gamma(1-\epsilon)}\left(\frac{4\pi\mu^2}{2p_i\cdot p_k}\right)^{\epsilon} = \frac{\alpha}{2\pi}\left[1+\left(\log\left(\frac{4\pi\mu^2}{2p_i\cdot p_k}\right)-\gamma\right)\epsilon\right]+O\left(\epsilon^2\right). \tag{4.39}$$

The factors of 4π and γ can be avoided by using $\overline{MS}$ scheme which means replacing

$$\mu'^2 = \frac{\mu^2}{4\pi}e^{\gamma}. \tag{4.40}$$

However as $\mathcal{V}_i(\epsilon)$ is usually at least as singular as $1/\epsilon$ this introduces an artificial dependence of the renormalization scale μ in the finite terms that have to be cancelled by the virtual corrections. The first implementation does exactly this, expanding every factor in (4.37) while using the $\overline{MS}$ scheme.

Another way to tackle this problem is to start with the definition of an insertion operator $\mathbf{I}(\epsilon)$ in such a way that the integration of a dipole can be written as (Eq.7.23 CS)

$$\int_{m+1} d\sigma^A = \int_m \left[d\sigma^B\cdot \boldsymbol{I}(\epsilon)\right]\ . \tag{4.41}$$

As shown in Eq.7.26 CS this insertion operator in the m-particle phase space is given by

$$\boldsymbol{I}(p_1,...,p_m;\epsilon) = -\frac{\alpha_\mathrm{S}}{2\pi}\frac{1}{\Gamma(1-\epsilon)}\sum_i\ \frac{1}{\boldsymbol{T}_i^2}\ \mathcal{V}_i(\epsilon)\ \sum_{k\neq i}\boldsymbol{T}_i\cdot\boldsymbol{T}_k\ \left(\frac{4\pi\mu^2}{2p_i\cdot p_k}\right)^{\epsilon}\ . \tag{4.42}$$

This insertion operator contains all the singularities and also finite terms. It is now useful to split this insertion operator into a part that contains the singularities and a part that contains the finite parts. We therefore define

$$\mathbf{I}(\epsilon) = I^{(1)}(\epsilon) + I_{fin}\ . \tag{4.43}$$

The $I^{(1)}(\epsilon)$-operator is defined in such a way that adding its contribution to the virtual corrections leads to a finite result. This requirement is however not sufficient to determine this operator in a unique way and there are various versions in the literature (see for instance [12], [13], [97], [98]).
We define our $I^{(1)}(\epsilon)$-operator as

$$I^{(1)}(\epsilon) = -\frac{\alpha_{\mathrm{S}}}{2\pi}\Gamma(1+\epsilon)\sum_i \frac{1}{\boldsymbol{T}_i^2}\,\mathcal{V}_i^{(1)}(\epsilon)\,\sum_{k\neq i}\boldsymbol{T}_i\cdot\boldsymbol{T}_k\,\left(\frac{4\pi\mu^2}{2p_i\cdot p_k}\right)^{\epsilon}\quad . \tag{4.44}$$

The factor $\mathcal{V}_i^{(1)}(\epsilon)$ is defined to contain only singular terms. In the massless case in the CS formulation, this factor is given by

$$\mathcal{V}_i^{(1)}(\epsilon) = \frac{\boldsymbol{T}_i^2}{\epsilon^2} + \frac{\gamma_i}{\epsilon}\,, \tag{4.45}$$

where γ_i is defined in Eq.(5.90) CS. In the case of γ_g there is one difference to the CS formulation. According to Eq.(5.90) CS this factor is

$$\gamma_g = \frac{11}{6}C_{\mathrm{A}} - \frac{2}{3}T_{\mathrm{R}}N_F\quad . \tag{4.46}$$

It contains two contributions coming from the splittings $g \to g\ g$ and $g \to q\ \bar{q}$ respectively. But as we start with one specific real emission process, we always know whether there was a splitting into two gluons or a splitting into a quark pair. So in our case we only have either one or the other contribution.
To replace the factor $\frac{1}{\Gamma(1-\epsilon)}$ by a factor $\Gamma(1+\epsilon)$ has no influence on the singular terms as one can write

$$\frac{1}{\Gamma(1-\epsilon)} = \Gamma(1+\epsilon) - \frac{\pi^2}{6}\epsilon^2 + \mathcal{O}(\epsilon^3)\quad . \tag{4.47}$$

This will however affect the finite terms. To see which terms are given by our second implementation we write it down explicitly.
The integrated splitting function $\mathcal{V}_i(\epsilon)$ contains in the most general case terms that are divergent as $1/\epsilon^2$, terms that are divergent as $1/\epsilon$ and finite terms. We can therefore write in a symbolic way

$$\mathcal{V}_i(\epsilon) = \frac{x_1}{\epsilon^2} + \frac{x_2}{\epsilon} + x_3\quad , \tag{4.48}$$

where x_1, x_2 and x_3 are again coefficients or functions, depending on the specific case one is considering. As $\mathcal{V}_i(\epsilon)$ is not more divergent as $1/\epsilon^2$ it is sufficient to consider only terms up to $\mathcal{O}(\epsilon^2)$ when replacing $1/\Gamma(1-\epsilon)$ by $\Gamma(1+\epsilon)$. We can therefore

write

$$\begin{aligned}\int [\mathrm{d}p_i(\widetilde{p}_{ij},\widetilde{p}_k)]\,\frac{1}{(p_i+p_j)^2-m_{ij}^2}\,\langle \mathbf{V}_{ij,k}\rangle \;&\equiv\; \frac{\alpha_\mathrm{s}}{2\pi}\frac{1}{\Gamma(1-\epsilon)}\left(\frac{4\pi\mu^2}{2p_i\cdot p_k}\right)^{\epsilon}\mathcal{V}_i(\epsilon)\\ &= -\frac{\alpha_\mathrm{S}}{2\pi}\left(\Gamma(1+\epsilon)-\frac{\pi^2}{6}\epsilon^2\right)\left(\frac{4\pi\mu^2}{2p_i\cdot p_k}\right)^{\epsilon}\left(\frac{x_1}{\epsilon^2}+\frac{x_2}{\epsilon}+x_3\right)\\ &= -\frac{\alpha_\mathrm{S}}{2\pi}\Gamma(1+\epsilon)\left(\frac{4\pi\mu^2}{2p_i\cdot p_k}\right)^{\epsilon}\left(\frac{x_1}{\epsilon^2}+\frac{x_2}{\epsilon}+x_3\right)\\ &\quad -\frac{\alpha_\mathrm{S}}{2\pi}\Gamma(1+\epsilon)\left(\frac{4\pi\mu^2}{2p_i\cdot p_k}\right)^{\epsilon}\underbrace{\frac{1}{\Gamma(1+\epsilon)}}_{1+\gamma\epsilon+\mathcal{O}(\epsilon^2)}\left(-\frac{\pi^2}{6}\epsilon^2\right)\left(\frac{x_1}{\epsilon^2}+\frac{x_2}{\epsilon}+x_3\right)\;.\end{aligned} \tag{4.49}$$

From the equation above one can see, that the introduction of $\Gamma(1+\epsilon)$ produces a contribution of the form $-\frac{\pi^2}{6}x_1$ to the finite terms.
When dealing with initial state hadrons there is another contribution coming from the collinear counterterms (Eq.(6.5) CS) and its contribution to the cross section is given by (Eq.(6.6) CS)

$$d\sigma_a^C(p;\mu_F^2) = -\frac{\alpha_\mathrm{S}}{2\pi}\,\frac{1}{\Gamma(1-\epsilon)}\sum_b\int_0^1 dz\,\left[-\frac{1}{\epsilon}\left(\frac{4\pi\mu^2}{\mu_F^2}\right)^{\epsilon}P^{ab}(z)+K^{ab}_{\overline{\mathrm{MS}}}(z)\right]\,d\sigma_b^B(zp)\;, \tag{4.50}$$

where in $\overline{MS}$ scheme, $K^{ab}_{\overline{\mathrm{MS}}}(z)=0$.
Neglecting the sum over the hadrons we have a contribution to the integrated dipole of the form

$$I_c(\epsilon) = -\frac{\alpha_\mathrm{S}}{2\pi}\frac{1}{\Gamma(1-\epsilon)}\left(\frac{4\pi\mu^2}{\mu_F^2}\right)^{\epsilon}\left[-\frac{1}{\epsilon}P^{ab}(z)\right]\quad. \tag{4.51}$$

Choosing our first implementation means expanding all factors in (4.51), i.e. we write

$$I_c(\epsilon) = \frac{y_{c,2}}{\epsilon}+y_{c,3}\;. \tag{4.52}$$

In the $\overline{MS}$ scheme the coefficients $y_{c,2}$ and $y_{c,3}$ are given by

$$\begin{aligned} y_{c,2} &= \frac{\alpha_s}{2\pi}\cdot P^{ab}(z)\\ y_{c,3} &= \frac{\alpha_s}{2\pi}\cdot P^{ab}(z)\log\left(\frac{\mu^2}{\mu_F^2}\right)\;.\end{aligned} \tag{4.53}$$

Choosing the second implementation first requires the replacement of $1/\Gamma(1-\epsilon)$. As the collinear counterterm is only proportional to $1/\epsilon$, this has no impact and we

can write

$$\begin{aligned} I_c(\epsilon) &= -\frac{\alpha_S}{2\pi}\Gamma(1+\epsilon)\left(\frac{4\pi\mu^2}{\mu_F^2}\right)^\epsilon \left[-\frac{1}{\epsilon}P^{ab}(z)\right] \\ &= -\frac{\alpha_S}{2\pi}\Gamma(1+\epsilon)\left(\frac{4\pi\mu^2}{2p_i\cdot p_k}\right)^\epsilon \underbrace{\left(\frac{2p_i\cdot p_k}{\mu_F^2}\right)^\epsilon}_{1+\log\left(\frac{2p_i\cdot p_k}{\mu_F^2}\right)\epsilon+\mathcal{O}(\epsilon^2)} \left[-\frac{1}{\epsilon}P^{ab}(z)\right] \\ &= -\frac{\alpha_S}{2\pi}\Gamma(1+\epsilon)\left(\frac{4\pi\mu^2}{2p_i\cdot p_k}\right)^\epsilon \left(\frac{x_{c,2}}{\epsilon}+x_{c,3}\right) \quad . \end{aligned} \tag{4.54}$$

From this we can immediately see that

$$\begin{aligned} x_{c,2} &= -P^{ab}(z) \\ x_{c,3} &= \log\left(\frac{\mu_F^2}{2p_i\cdot p_k}\right)P^{ab}(z) \quad . \end{aligned} \tag{4.55}$$

The final output is then given by all contributions of the coefficients x_i and y_i respectively multiplied with a born level matrixelement that is modified by its color structure.
For the first implementation this explicitly means

$$\begin{aligned} &\tfrac{1}{\epsilon^2}: && y_1 \cdot {}_m\langle 1\cdots m|\frac{\boldsymbol{T}_i\cdot\boldsymbol{T}_k}{\boldsymbol{T}_i^2}|1\cdots m\rangle_m \\ &\tfrac{1}{\epsilon}: && (y_2+y_{c,2})\cdot {}_m\langle 1\cdots m|\frac{\boldsymbol{T}_i\cdot\boldsymbol{T}_k}{\boldsymbol{T}_i^2}|1\cdots m\rangle_m \\ &\text{finite:} && (y_3+y_{c,3})\cdot {}_m\langle 1\cdots m|\frac{\boldsymbol{T}_i\cdot\boldsymbol{T}_k}{\boldsymbol{T}_i^2}|1\cdots m\rangle_m \quad , \end{aligned}$$

and for the second implementation we find

$$\begin{aligned} &\tfrac{1}{\epsilon^2}: && x_1 \cdot {}_m\langle 1\cdots m|\frac{\boldsymbol{T}_i\cdot\boldsymbol{T}_k}{\boldsymbol{T}_i^2}|1\cdots m\rangle_m \\ &\tfrac{1}{\epsilon}: && (x_2+x_{c,2})\cdot {}_m\langle 1\cdots m|\frac{\boldsymbol{T}_i\cdot\boldsymbol{T}_k}{\boldsymbol{T}_i^2}|1\cdots m\rangle_m \\ &\text{finite:} && (x_3-\tfrac{\pi^2}{6}x_1+x_{c,3})\cdot {}_m\langle 1\cdots m|\frac{\boldsymbol{T}_i\cdot\boldsymbol{T}_k}{\boldsymbol{T}_i^2}|1\cdots m\rangle_m \quad . \end{aligned}$$

The contributions from the collinear counterterms are of course only present if there are initial state QCD particles involved.
The explicit values of x_1, x_2 and x_3 are given by Eq.(4.45) and Eq.(7.27) CS. They can be expressed in a general way,namely

$$\begin{aligned} x_1 &= \boldsymbol{T}_i^2 \\ x_2 &= \gamma_i \\ x_3 &= -\boldsymbol{T}_i^2\frac{\pi^2}{3}+\gamma_i+K_i \, , \end{aligned} \tag{4.56}$$

where K_i is defined in Eq.(7.28) CS. According to our previous considerations the values of y_1 and y_2 are given by

$$\begin{aligned} y_1 &= -\frac{\alpha_s}{2\pi}\boldsymbol{T}_i^2 \\ y_2 &= -\frac{\alpha_s}{2\pi}\left(\gamma_I + \log\left(\frac{\mu^2}{2p_i\cdot p_k}\right)\right) \\ y_3 &= -\frac{\alpha_s}{2\pi}\left(\gamma_i \log\left(\frac{\mu^2}{\mu_F^2}\right) + \gamma_i + K_i + \frac{1}{2}\boldsymbol{T}_i^2 \log^2\left(\frac{\mu^2}{\mu_F^2}\right) - \frac{5\pi^2\boldsymbol{T}_i^2}{12}\right) . \end{aligned} \tag{4.57}$$

4.3 Check of the α-parameter implementation

Both the not-integrated and the integrated dipoles depend on the α-parameter. The result of a full NLO calculation must of course be independent of the actual value of this parameter. It is therefore very important to check these dependencies and proof the independence of the full result. We require that

$$\int_{n+1}\left(d\sigma^R - d\sigma^A\right) + \int_n (\text{finite parts of int. dip.}) = \text{const}, \tag{4.58}$$

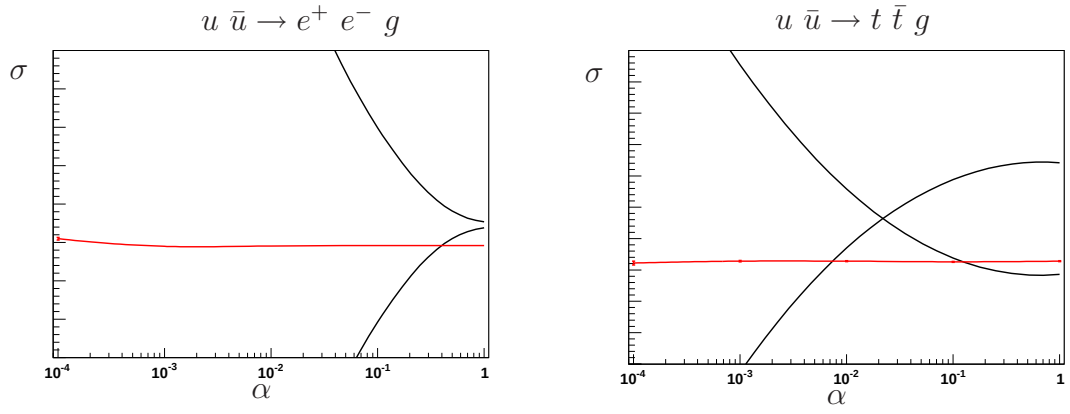
which must be a constant in that sense that it is not a function of α. This has to be proven for all the 27 different combinations listed above.

Figure 1 shows the dependence on the α-parameter for two examples. The example on the left side shows an initial-initial dipole and the picture on the right side shows the initial-final dipole. In this case the spectators are the massive top quarks. The red lines show the quantity defined in (4.58) and confirm the requirement that this quantity must be independent of α to a very good precision over several orders of magnitude. For the two contributions (black lines) we neglect the error bars for simplicity, they are however taken into account in the sum (red lines). The unit of the result is of course the unit of a total cross section. As our result is only valid up to an overall normalization (which is absolutely sufficient for this test) we do not give a precise result here.

4.4 Regularization scheme dependence

Calculating processes that suffer from divergencies requires a systematic prescription of how to deal with these divergencies. The full NLO result has to be independent of such a regularization scheme, but the real emission part and the virtual corrections do depend on the scheme only the sum does not. Therefore it is necessary to clearly specify which regularization scheme one is using.

There are mainly two types of regularization schemes used, namely dimensional regularization [1,99–101] and dimensional reduction [102–104]. A very useful discussion



(a) (b)

Figure 4.1: Two examples for the α-dependence is shown. The picture on the left shows the initial-initial dipole where a gluon is radiated of the initial quark lines, the second example is an initial-final dipole where the spectator is massive. In both cases the increasing black line is the first contribution of (4.58), i.e. the sum of the matrix element plus the non integrated dipoles. The decreasing black line is the second contribution, i.e. the finite terms of the integrated dipoles. The red line is the sum of the two black lines. The x-axis denotes the range over which α is varied, the y-axis denotes the total cross section up to an overall normalization so that we do not give a specific range here.

about their subtypes and their differences can be found in [105].
The basis on which MadGraph and also MadDipole build up the amplitudes are the HELAS routines [85]. They evaluate the matrix element in four dimensions. The integration over the one particle phase space is divergent and this divergence is regularized using dimensional regularization. Keeping all external and internal particles four dimensional but shifting the integration variables to d dimensions is exactly the 't Hooft Veltman scheme (tHV). For our purposes this is in some sense the most natural regularization scheme. However dimensional reduction is a widely used regularization scheme. As both methods differ just in finite shifts we implemented both. To be more precise: As we treat external gluons four dimensional it is rather the four-dimensional helicity scheme (FDH) than the original dimensional reduction (DRED) which treats external gluons as quasi four dimensional.
According to [12, 106], going from the 't Hooft Veltman scheme to the four dimensional helicity scheme requires the following replacement:

$$\mathcal{V}_I(\epsilon)^{tHV} \to \mathcal{V}_I^{\text{FDH}}(\epsilon) = \mathcal{V}_I(\epsilon)^{tHV} - \tilde{\gamma}_I + \mathcal{O}(\epsilon) \ , \tag{4.59}$$

$$\tilde{\gamma}_q = \tilde{\gamma}_{\bar{q}} = \frac{1}{2} C_F \ , \qquad \tilde{\gamma}_g = \frac{1}{6} C_A \ . \tag{4.60}$$

In the massive case there is no dependence on the regularization scheme [98].

4.5 Implementation of the integrated dipoles

In addition to the already existing MadDipole package, the inclusion of the integrated dipoles delivers one additional fortran subroutine. It takes the form

`intdipoles(P, Z, EPSSQ, EPS, FINITE)`,

where the input parameters are the phase space point `P(0:3,nexternal)` and `Z`. `Z` is the momentum fraction of the incoming parton that goes into the hard process after an initial state radiation or if the spectator is an initial state particle. It is exactly the same quantity which is in CS denoted with $x_{ij,a}$, $x_{ik,a}$ and $x_{i,ab}$ respectively.
The output parameters are the the coefficients of the divergent and finite terms. After inclusion of the collinear counterterms the divergent terms only occur at $z = 1$. Therefore these parameters are just given as real numbers. This is however not the case for the finite terms. When dealing with initial state partons, the integrated splitting function contains distributions. We therefore write the finite terms as a 5-dimensional array. The first entry contains the terms which are evaluated at $z = 1$ which means they are proportional to $\delta(1-z)$. The second entry contains the terms that are a regular function in z, i.e. they are finite for any value of z in the range between 0 and 1. The +-distribution terms are split between the second and the third entry. This is done in the following way. If we have a +-distribution acting on a test function $g(x)$ we divide

$$\int dx \left(\frac{1}{1-x}\right)_+ g(x) = \int dx \underbrace{\frac{g(x)}{1-x}}_{\text{part of the second entry}} - \underbrace{\frac{g(1)}{1-x}}_{\text{part of the third entry}} . \tag{4.61}$$

In the massive case the point $z = 1$ can not be reached in all cases but we may have a reduced endpoint which is given by

$$x_+ = 1 - 4\mu_Q^2, \tag{4.62}$$

where μ_Q is the rescaled fermion mass occurring in the splitting.
Instead of a $\delta(1-x)$ we then have a δ-distribution of the form $\delta(x_+ - x)$ which is the fourth entry of the array of the finite terms. If we have such a reduced endpoint then also the +-distribution is generalized into a x_+-distribution as defined in (4.28). These contributions are stored in the fifth entry. In the same way as for the +-distribution (4.61), the terms from the x_+-distributions are split up into two parts, one is stored in the second entry, the other in the fifth. Writing this out explicitly as in (4.61) means

$$\int dx \left(\frac{1}{1-x}\right)_{x_+} g(x) = \int dx \underbrace{\frac{g(x)}{1-x}}_{\text{part of the second entry}} - \underbrace{\frac{g(x_+)}{1-x}}_{\text{part of the fifth entry}} . \tag{4.63}$$

Chapter 5

How to use the package

The installation and running of the MadDipole package is very similar to the usual Stand-Alone version of the MadGraph code. In this section, we only provide a brief description for MadDipole, while more information can also be found on the MadGraph wiki page,
`http://cp3wks05.fynu.ucl.ac.be/twiki/bin/view/Software/MadDipole`.

1. Download and extract the MadDipole package, `MG_ME_DIP_V4.4.3.tar.gz`, from one of the MadGraph websites, *e.g.*, `http://madgraph.hep.uiuc.edu/`.

2. Run `make` in the `MadGraphII` directory.

3. Copy the `Template` directory into a new directory, *e.g.*, `MyProcDir` to ensure that you always have a clean copy of the Template directory.

4. Go to the new `MyProcDir` directory and specify your process in the file `./Cards/proc_card.dat`. This is the $(n+1)$-particle process you require the subtraction term for.

5. Running `./bin/new_process` generates the code for the $(n+1)$-particle matrix element and for all dipole terms. After running this you will find a newly generated directory `./Subprocesses/P0_yourprocess` (*e.g.*, `./Subprocesses/P0_e+e-_uuxg`) which contains all required files.

6. For running the check program change to this directory and run `make` and `./check`.

The directory `./Subprocesses/P0_yourprocess` contains all necessary files needed for further calculations. As in the usual MadGraph code the $(n+1)$-particle matrix element is included in the file `matrix.f`. For the dipoles there exist several files called `dipol???.f` where `???` stands for a number starting from `001`. Each file contains only one dipole. Note that the syntax for calling the dipoles is exactly the same as calling the $(n+1)$-particle matrix element. In particular, also the dipoles

have to be called with $(n+1)$ momenta rather than with only n momenta. The remapping of the momenta going from the $(n+1)$-particle phase space to a n-parton phase space is done within the code for the dipoles.
The file `dipolsum.f` calculates the sum over all dipoles for a given $(n+1)$-particle phase space point. It contains two subroutines called `DIPOLSUM` and `DIPOLSUMFINITE`. As discussed in Section 3.4 above, the subroutine `DIPOLSUMFINITE` contains the dipoles which only contribute potentially large logarithms but not a real singularity. In both subroutines the contribution of a certain dipole is only added to the sum if the phase space restriction specified by the α parameter is fulfilled. The value of the four α parameters can be changed in these two subroutines. They are all set to unity by default.
For the integrated dipoles the subroutine `intdipoles` is generated and written in the file `intdipoles.f`. The composition of this file is very similar to that in `dipolsum.f`. Within this subroutine the divergent and finite terms are calculated. The calculation of the prefactors of this term can be found in the files `epsterms.f` and `finiteterms.f` respectively.

Chapter 6

NLO QCD corrections to $q\bar{q} \to b\bar{b}b\bar{b}$

6.1 Motivation

The main motivation for the calculation of this process comes from beyond the Standard Model scenarios. One possibility is a supersymmetric extension of the Standard Model. Supersymmetric models introduce an additional bosonic field for each existing fermion field and a fermionic field for each existing bosonic field. One of the simplest examples for a supersymmetric model is the so called minimal supersymmetric extension of the Standard Model (MSSM). In order to maintain supersymmetry it turns out that it is necessary to introduce s second Higgs doublet. One of these Higgs doublets (Φ_u) couples to up-type quarks whereas the other couples to down-type quarks and is labelled with Φ_d. After symmetry breaking one ends up with five physical states. Two of them are neutral CP-even scalar fields called H_0 and h_0, one CP-odd field A_0 and two charged fields called H^+ and H^-.
One of the important characteristic quantities of supersymmetric models is the ratio of the vacuum expectation values of the Higgs fields which one writes as

$$\tan\beta = \frac{|\langle\Phi_u\rangle|}{|\langle\Phi_d\rangle|} \,. \tag{6.1}$$

The value of $\tan\beta$ is a priori not known but a free parameter of the theory that has to be determined by the experiment. For a large $\tan\beta$ the coupling $H_0 b\bar{b}$ is enhanced and therefore one expects to see signals of supersymmetric models in channels associated with b-quarks. For a detailed discussion see [107–109]. To distinguish supersymmetric signals from Standard Model background this background has to be known as good as possible and so the calculation of the next-to-leading order QCD corrections are an important step for a better understanding of the Standard Model background.

6.2 Real emission

The process contributing to the real emission part where the leading order process is $q\bar{q} \to b\bar{b}b\bar{b}$ is given by

$$q\bar{q} \to b\ \bar{b}\ b\ \bar{b}\ g. \tag{6.2}$$

Throughout the whole calculation we treat the b-quark as a massless particle. The calculation of the real emission matrix element and the subtraction terms was done by using the MadDipole package. For this process there exist 50 dipoles. However as we require that four b-jets have to be detected we can neglect the dipoles coming from the singularity where two b-quarks become collinear. This means that only 30 dipoles really have to be evaluated.
For the integration we use the standard MadEvent package. For simplicity we constructed the matrix element only for a u-type quark. As we have a pure QCD process here it is sufficient to just sum over the different parton distribution functions (pdf's) using the same matrix element. As contributions coming from initial state b's are much smaller compared to the other flavors they have been omitted
Concerning the cuts we had to modify the original MadEvent approach. The difference here is that one has to put a cut on the $n+1$ matrix element using the set of $n+1$ momenta given by the integration routine as well as putting a cut on the dipoles but using the remapped momenta according to the dipole prescription. This implies a different cut for each single dipole. For the $n+1$ matrix element it is a crucial question whether the additional gluon can be combined to a jet which means that further cuts are applied only on the combination of these particles. An appropriate treatment therefore requires the introduction of a jet algorithm.

6.2.1 Jet algorithm

For this calculation we used the $k_\perp$ algorithm described in [110]. For our case it is not only important to know whether two particles can be combined to a jet or not, but also to ensure that events where two b-quarks would be combined to a jet are discarded as well as events where more than two particles can be combined. Following the prescription of [110] we calculate the object

$$d_i = E_{T,i}^2 \tag{6.3}$$

for each particle, as well as

$$d_{ij} = min(E_{T,i}^2, E_{T,j}^2) \cdot \frac{\Delta R_{ij}^2}{R^2} \tag{6.4}$$

for each combination. Here E_T is the transverse energy and ΔR_{ij} specifies the (η, ϕ)-separation between two particles. The parameter R is a resolution parameter which should be chosen to be of order 1. We set it to the same value that we require in

the cut on the R-separation in the n particle phase space, namely $R = 0.8$.
We now sort all possible combinations of d_i's and d_{ij}'s in a list such that the first entry is the smallest. If this is a d_i this particle itself already forms a jet and is not further mergable. It is therefore removed from the list of particles and considered to be a jet. If the smallest entry is the combination of a b-quark and the gluon we combine these two particles by simply adding their momenta. This gives us a new set of n-momenta on which we then apply the same cuts that have to be fulfilled by the dipoles and the other n-particle contributions. If the smallest entry is given by a combination of two b-quarks the event is discarded. Otherwise these steps are repeated until all jets have been found.

6.3 Virtual corrections

For the virtual corrections we use the code that has been developed by the GOLEM collaboration [45, 46, 61, 111–114]. It is basically one subroutine that can be used as a blackbox. The only input parameters which are needed is the phase space point $P(0:3, nexternal)$ and the renormalisation scale μ_R. In return it gives a four dimensional array containing the value of the matrix element squared of the born process, the contribution coming from the virtual corrections and the coefficients of the $\frac{1}{\epsilon^2}$- and $\frac{1}{\epsilon}$-terms.
The GOLEM routine has been implemented to a MadEvent package that was originally designed for the calculation of the cross section for the leading-order process. In this package we implemented in addition the integrated dipoles and the routine for the virtual corrections. To include the virtual corrections and the integrated dipoles in one integration routine has the advantage that the coefficients of the divergent terms can be compared between the two parts for each single phase space point. This is another important check to ensure the correct implementation of the different parts.
For a correct treatment of the integrated dipoles it was necessary to extend the phase space by an additional variable that comes along with the integrated terms.
When doing the integration it turned out that there exist points in phase space where the ratio of virtual corrections compared to the leading order matrix element squared becomes unnaturally large, i.e. easily more than one order of magnitude larger than the leading order contribution. This behaviour of single points can completely destroy the convergence of the integration and is known [114]. To solve this problem we impose a cut on this ratio. We define it as

$$K_{cut} = \left| \frac{2\mathcal{M}^*_{virt}\mathcal{M}_{LO}}{|\mathcal{M}_{LO}|^2} \right| . \tag{6.5}$$

Varying the maximally allowed value for K_{cut} shows that there exists a plateau where the value for the integral is independent of the specific value of K_{cut}. In particular

it started to be independent for values larger than five so we set

$$K_{cut} = 5 \ . \tag{6.6}$$

Phase space points with larger values for K_{cut} have been discarded. However these events were written into an extra event file such that they can be recalculated with quadrupole precision to determine their numerical significance. We leave this issue for further checks.
Another possibility is to take an event sample with unweighted events based on the leading order process and do a reweighting according to the virtual contributions. This has the advantage that in the step of calculating the virtual corrections no Monte Carlo integration is needed. So the possibility that the Monte Carlo integration finds a large weight and adapts the grid around this possibly unnatural peak thereby spoiling the integration does not exist. Another advantage of this method is that this can parallelised very easily. This is of course also possible for a Monte Carlo integration. But assuming that one uses a single processor for one run this requires a minimal number of phase space points to make the integral converge. For this calculation one phase space point took approximately three seconds. With a minimal number of around 20 000 phase space points for this n-particle phase space, one run takes about one day on a modern CPU. And this restriction no longer holds when using a sample of unweighted events and reweighting it. Unfortunately we have to leave this for the future.

6.4 Numerical results

The calculation consists of two completely independent steps. One is the integration over the $n+1$-particle phase space integrating the matrix element squared of the real emission contribution where the dipoles are subtracted in order to get a finite result. The second step is the integration over the n-particle phase space where the integrand is given by the matrix element squared of the leading-order process, the integrated dipoles and the virtual corrections.
Throughout the whole calculation we use the following cuts for the b-jets:

$$\begin{aligned} P_T &\geq 30 \text{ GeV} \\ \eta &\leq 2.5 \\ \Delta R &\geq 0.8 \ . \end{aligned} \tag{6.7}$$

The transverse momentum P_T is defined as

$$P_T = \sqrt{P_x^2 + P_y^2} \ . \tag{6.8}$$

η denotes the pseudo-rapidity of a particle and is given by

$$\eta = -\ln\left(\tan\left(\frac{\theta}{2}\right)\right) \tag{6.9}$$

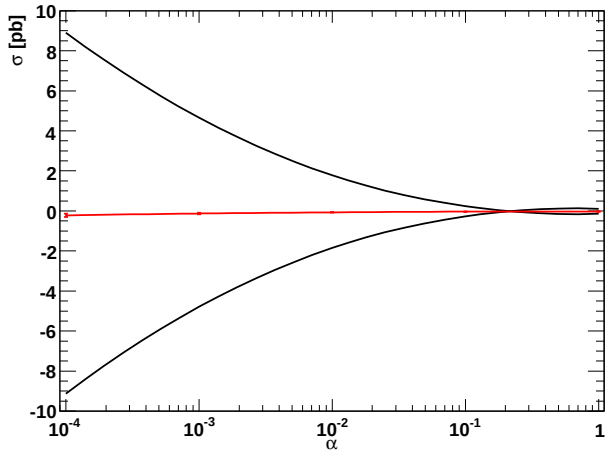


Figure 6.1: This figure shows the α dependence of the total result. The ascending line shows the real emission matrix element squared minus the dipoles, the descending line shows the integrated dipoles. The red line is the sum of both contributions and shows the independence of α to a very good approximation.

where θ is the angle of the particle relative to the beam axis. The R-separation between two particles i and j is defined by

$$\Delta R = \sqrt{(\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2} \ , \tag{6.10}$$

where ϕ is the azimuthal angle.

Furthermore we use the $\overline{MS}$ scheme as described in chapter four. For the parton distribution functions we used the CTEQ6M pdf's. The factorisation scale has been fixed to

$$\mu_F = 100 \text{ GeV} \ . \tag{6.11}$$

The reason why we keep the factorisation scale fixed is that we would need to include also the quark-gluon channel. Varying the factorisation scale means shifting terms between the parton distribution function and the matrix element and to do this in a consistent manner one needs to include these matrix elements.
A first check for the integration is the independence on the cut parameter α as described in chapter 4. Figure 6.1 shows the result. To a very good approximation the total result is independent on the cut parameter. This is a powerful check that ensures the correct settings of the integration. It also proves the sensibility of the integration of the real emission. Having a correct behaviour when varying the parameter shows that the integration routine is sensitive on that cut and does not return some arbitrary number. On the side of the integrated dipoles it also shows

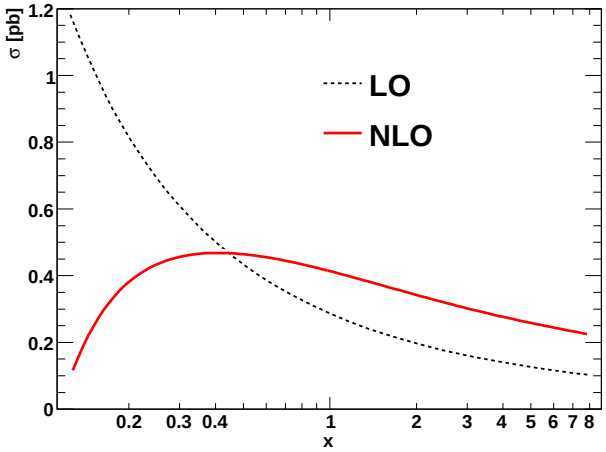


Figure 6.2: This figure shows the scale dependence of the LO and the NLO contribution. The factorisation scale is fixed to $\mu_F = 100\ GeV$ and the renormalisation scale is varied with $\mu_R = x\mu_0$.

the correct interplay between the terms proportional to the δ function and the terms being a regular function or a +-distribution. This implies that for instance the pdf's are called with their correct values. Any inconsistency here would immediately spoil the α independence.
The ingredients of the integrated dipoles have been compared with an implementation in WHIZARD [22]. We have compared all the different parts for single phase space points and also for different values of α and found full agreement [115].
For the following numerical results we choose

$$\alpha = 0.01\ . \tag{6.12}$$

Both large and small values for α have advantages and disadvantages. For very small values of α a slight deviation from the α independence can be observed. The advantage is that the integration converges better. Only near a singularity the dipoles are evaluated. This means that one is basically integrating only the real emission matrix element throughout the largest part of the phase space. And as MadEvent is optimized for exactly this it is no surprise that the integration works better. For large values of alpha the integration becomes worse and also the size of the event files increases strongly as each dipole gives one event. So the more dipoles are calculated the more events are written.

One important result in the calculation is the behaviour of the scale dependence. This result is shown in Figure 6.2. For the scale variation the factorisation scale is fixed according to (6.11). The renormalisation scale is varied around a central value

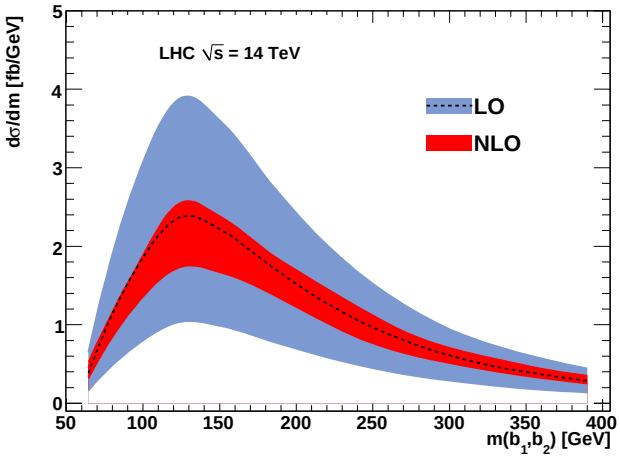


Figure 6.3: This figure shows the uncertainty for the invariant mass of the two b-jets with the highest transverse momentum. The blue shaded area is the covered range when varying the renormalisation scale for the leading order contribution from $2\mu_0$ to $\mu_0/4$. The red shaded area shows the same variation for the next-to-leading order result and the dashed line denotes the leading order result for the value $\mu_0/2$. As expected a clear reduction on the scale dependence can be observed.

μ_0

$$\mu_R = x\mu_0 \ , \tag{6.13}$$

with

$$\frac{1}{8} \leq x \leq 8 \ . \tag{6.14}$$

We choose the central value μ_0 to be the geometric sum of the transverse momenta of the particles:

$$\mu_0 = \sqrt{\sum_i P_{T,i}^2} \ . \tag{6.15}$$

Figure 6.2 shows the typical reduction of the scale dependence occurring when taking into account the next-to-leading-order corrections. For a value where the renormalisation scale is around half of μ_0 leading order and next-to-leading order give the same contribution. As an example how the scale variation effects the observable we show the invariant mass distribution for the two b-jets with the highest transverse momenta for a variation of the renormalisation scale from $\mu_0/4$ to $2\mu_0$. The result is given in figure 6.3. When varying the renormalisation scale the result for the leading order contribution lies within the blue shaded region. Doing the same variation for the next-to-leading order result one sees that the dependence on the scale is clearly reduced. The variation then lies within the red shaded region. From Figure 6.2 one

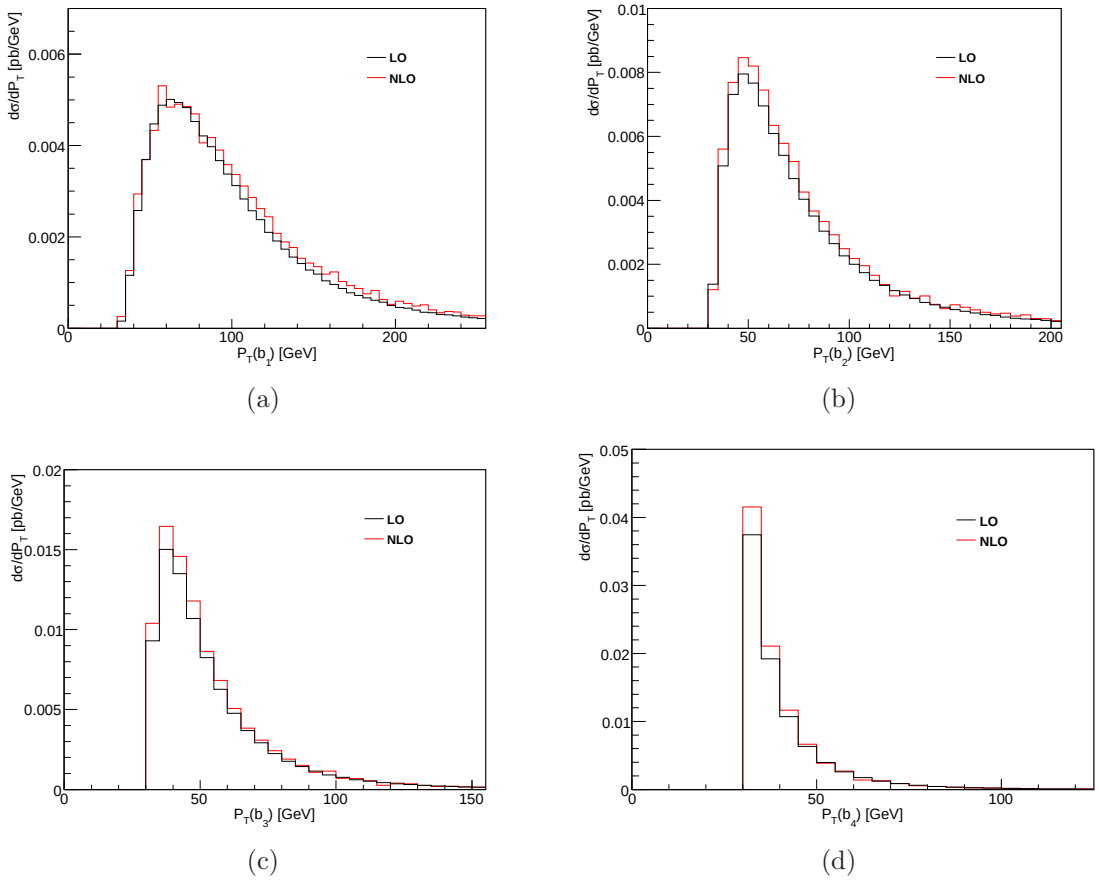


Figure 6.4: This figure shows the distribution of the transverse momentum P_T of the b-jets. b_1 denotes the b-jet with the highest transverse momentum, b_4 the one with the lowest momentum. The distribution is shown for both leading order (black) and next-to-leading order (red). A cut of $P_T \geq 30$GeV has been put on the quarks.

can see that for scales around

$$\mu_r = \frac{1}{2}\mu_0 \ , \tag{6.16}$$

the total cross section for leading order and next-to-leading order are very similar. For a better comparison between leading order and next-to-leading order in the various distributions we choose this scale for all the following plots.

For all the following distributions we have gathered $3 \cdot 10^8$ phase space points for the integration over the $n+1$-particle phase space. This large number becomes necessary not only because of the large dimensionality of the phase space but in particular due to the effect of 'missed binning'. This means that there are points in phase space in particular near a singularity, where both matrix element and subtraction terms come along with a large weight. The matrix element has a positive

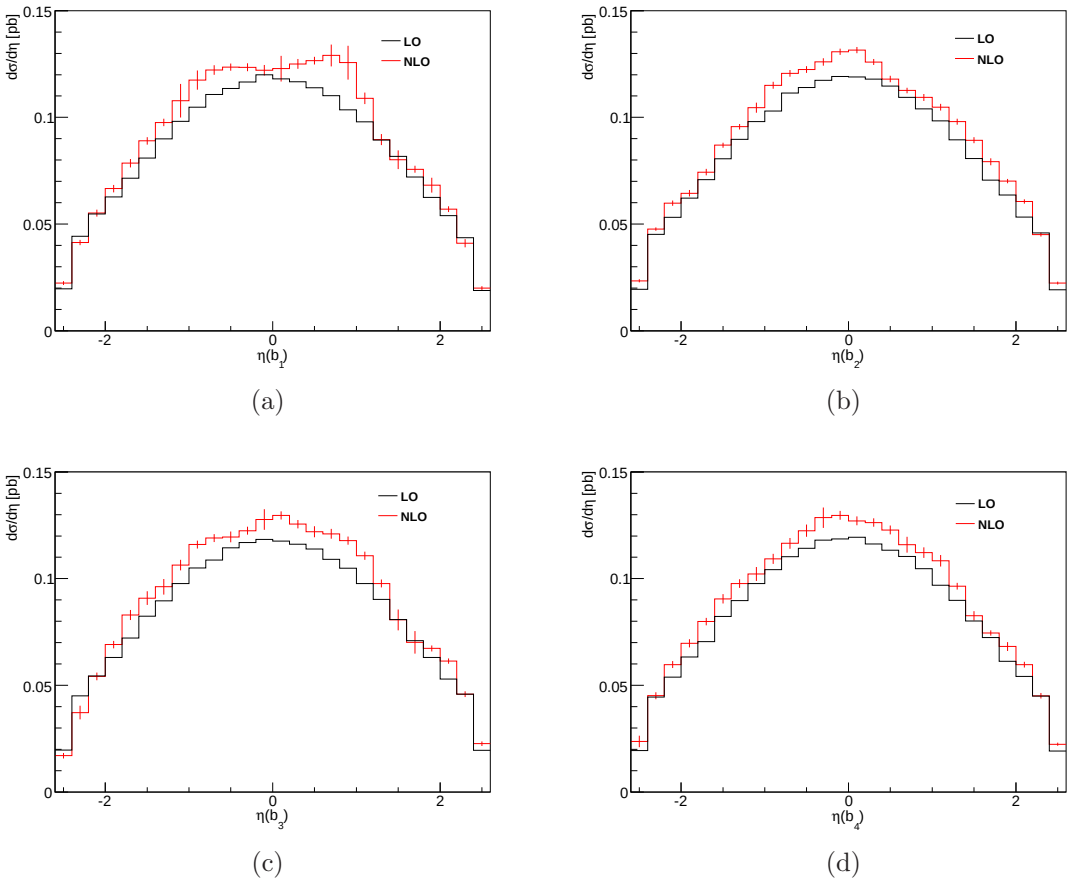
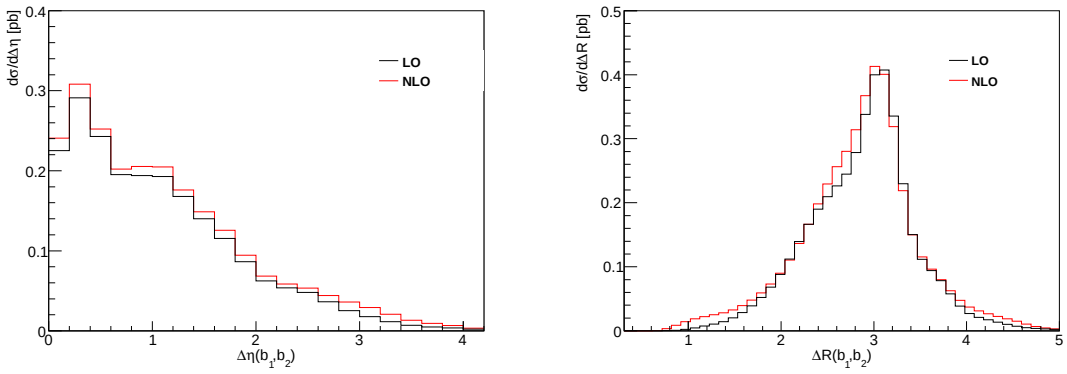


Figure 6.5: This figure shows the distribution of the pseudo-rapidity η of the b-jets. Again the quarks are ordered according to their transverse momentum P_T. As we have imposed a cut such that $|\eta| \leq 2.5$ we have limited the plotting range to this region.

weight whereas the subtraction terms come with a negative weight. There is now the possibility that they do not end up in the same bin but in bins next to each other. This leads to the effect that there is a peak in one bin and a dip in the next bin. To flatten out these missed binning events a lot of statistics is required.
For the integration over the n-particle phase space we used $1.2 \cdot 10^6$ phase space points. This gave roughly the same quality of the distributions as for the $n+1$ part. The reason why we did not choose a higher number was because the computing time with about three seconds per phase space point was so much larger than for the real emission part. The computing time for the $n+1$-particle integration strongly depends on the fact whether the histograms are booked 'on the fly' or whether the event is written in an event file. So the required time for one phase space point varied between 1.5ms and 5ms.



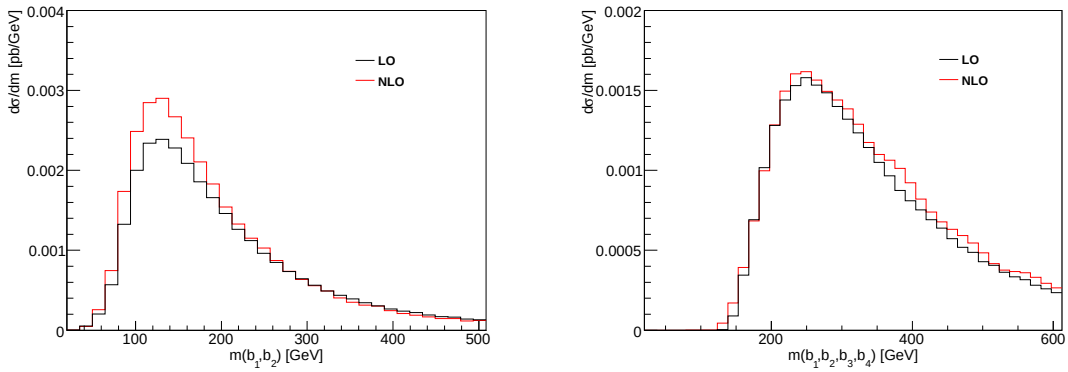
(a) (b)

Figure 6.6: Plot (a) shows the difference in the pseudo-rapidity for the two b-jets with the highest transverse momentum. In plot (b) the difference in the R-separation of these two quarks is drawn.

One important distribution is the distribution of the transverse momentum P_T of the b-jets. As they are identical particles we order them according to their transverse momentum. With b_1 we denote the b-jet with the highest transverse momentum followed by b_2 and so on. The plots of this observable are shown in figure 6.4. As expected from the choice of the scale the difference between the leading order result and the next-to-leading order result is small.
In figure 6.4 we plotted the pseudo-rapidity η of the four quarks. The quarks are again ordered according to their transverse momentum P_T such that $\eta(b_1)$ shows the rapidity of the quark with the largest P_T. As one can still see some statistical fluctuations we also printed the error bars for the NLO part. With the error bars one can see that the occurring little peaks leading to a slightly asymmetric shape also come with a larger error and therefore do not have a physical meaning.

A similar quantity as the pseudo-rapidity itself is the difference in the pseudo-rapidity between two quarks. This is shown in part (a) of figure 6.6 for the case with the two quarks with the highest transverse momentum.
In part (b) of that figure the difference in the pseudo-rapidity is combined with the difference in the azimuthal angle to the R-separation as defined in 6.10. This plot nicely shows that the next-to leading order correction does not change the geometrical alignment of the two hardest b-jets.

A very important observable in view of new particle searches are the invariant masses of the combination of particles. The invariant mass of two particles is defined by



(a) (b)

Figure 6.7: Invariant mass distribution of the two hardest b-jets (a) all four b-jets (b).

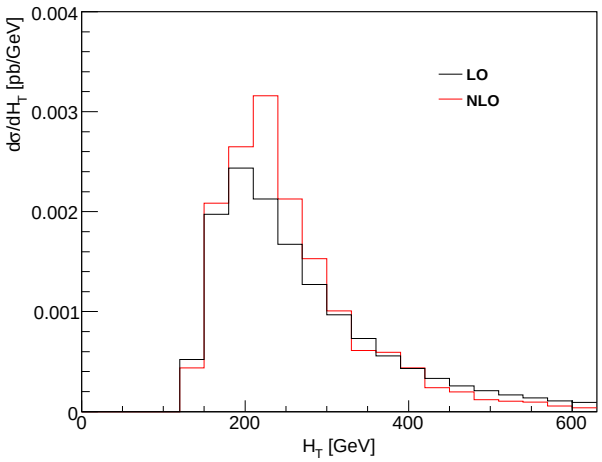


Figure 6.8: In this plot the distribution of the sum of the transverse energies H_T is shown. Here one can see a sizeable difference between the LO and the NLO result.

the four dimensional scalar product of the sum of the momenta,

$$m_{ij} = \sqrt{(P_i + P_j)^2} \ . \tag{6.17}$$

In figure 6.7 the invariant mass of the two hardest b-jets is drawn as well as the combination of all four b-jets. In figure 6.4 we show the H_T distribution of the final states particles. The observable H_T is defined as the sum of the transverse energies,

$$H_T = \sum_i E_{T,i} \ . \tag{6.18}$$

In the case of massless particles this corresponds to the sum of the transverse momenta. For this observable we see a noticeable difference between the leading order and the next to leading order result as the peak is higher and shifted slightly towards higher energy.

6.5 Outlook

In this chapter we presented the next-to-leading order corrections in QCD to the process $q\bar{q} \to b\bar{b}b\bar{b}$. For a full realistic description of LHC physics one also has to take into account the channel $gg \to b\bar{b}b\bar{b}$. This channel is more intricate than the quark channel due to the more complicated color structure of its amplitudes. That is why it has 90 dipoles in total compared to the 50 in the quark channel. As there are no more additional singularities we do not expect the integration of the real emission to be more complicated we rather expect that only the time per phase space point increases. First tests have confirmed these considerations. The implementation into already existing MadEvent packages is then identical as in the quark channel.
For completeness we also mention that there is also the channel with a quark and a gluon in the initial state. This has of course also to be taken into account. There are however no new virtual contributions associated to this channel.
First tests concerning the integration of the gluon channel have been successful so we are optimistic that we will be able to present the full result in the nearest future.

Chapter 7

Conclusions

There's a long tradition in theoretical physics, which by no means affected everyone but certainly affected me, that said the strong interactions are too complicated for the human mind.
–S. Weinberg

The Standard Model of particle physics has been a very successful theory of the fundamental interactions in nature except gravity. However there are various hints and reasons to believe that the Standard Model cannot be the complete theory. First of all it cannot describe gravity on the level of a quantum theory. Furthermore it does not give an explanation for dark matter and dark energy. But as the Standard has been consistent with all experiments to a very good approximation the only chance to discover new physics with today's experiments lies in better and more precise measurements and in better and more precise theoretical calculations.
From a perturbation theoretical point of view this means that theoretical results have to be known not only at leading order in perturbation theory but also at next-to-leading order at least for important processes. These processes are listed and updated in the Les Houches 'wishlist' [116]. For some processes even a result at next-to-next-to-leading order would be desirable.
Any of such calculations requires a lot of effort in every respect. New techniques and modern computing facilities allow to tackle problems that seemed not be feasible some years ago. The first full NLO QCD corrections to $2 \to 4$ processes have become available recently [62, 117, 118].

One main ingredient of an NLO calculation is the question how to deal with infrared divergencies. The most common approach to solve this problem is the introduction of subtraction terms which cancel the infrared divergencies in the real emission part and in the virtual part separately. To construct and calculate these subtraction terms is an intricate and time consuming part of the full calculation.
In this thesis we presented how the construction of subtractions terms for next-to-leading order in QCD can be done in a fully automated way for an arbitrary process.

For the construction we followed an approach proposed by Catani and Seymour. To solve this problem required basically two steps: The construction of subtraction terms for the real emission process and on the other hand the integrated subtractions terms for the virtual corrections. In chapter three and four we described the way of automation in detail. Several important checks have been made in order to ensure correctness of the result. The first part, the subtraction terms for the real emission part has already become publicly available and has already been used in actual calculations (see e.g. [96, 119]). The second part, the inclusion of the integrated terms is not yet public but will be published soon.
The automation was done for both massless and massive cases hence giving the user a very general tool for various purposes.
In the fifth chapter we give a short description how the software package can be installed and set up.

As an application for the automated subtraction terms we calculated the next-to-leading order corrections for the production of four b-quarks for the quark channel. The production of four b-quarks is an important channel for searches beyond the Standard Model especially in supersymmetric extensions. Here the coupling of Higgs bosons to b-quarks can be strongly enhanced compared to the Standard Model contribution. The Standard Model calculation is therefore important for a precise determination of the background. The details of this calculation and the results have been discussed in chapter six. For the calculation we have combined our MadDipole package with the publicly available MadEvent package as the main framework. In this framework we have built in an external routine from the GOLEM collaboration to include the virtual corrections.
This is one example how calculations could be done in the future. Instead of repeating the same steps for each calculation and where each collaboration develops basically the same things independently, it could be very convenient to use different modules for specific purposes and combining them according to the desired goal. This of course requires an agreement on conventions and on the exact format of in- and output data of the single modules. Although not yet official there will be such an agreement in form of a Les Houches accord the same way as has been for the standard output of event files [87].

As we have only included initial state quarks in our calculation we still need the inclusion of the mixed quark-gluon channel and the gluon-gluon channel. As already discussed in chapter six the main difference there is the more complicated color structure of the matrix elements hence leading to larger expressions which comes along with an increase of the required computing time. However and more important, as there are no additional singularities occurring we do not expect further complications as in the quark channel. First tests of the integration confirm this claim.

Another natural extension of this automated subtraction procedure would be the inclusion of electroweak corrections where a photon becomes soft or collinear to a fermion. The conceptual ideas of this procedure have been explained in [84,120]. There are two main differences compared to the QCD case: Traditionally electroweak calculations are done using mass regularisation instead of dimensional regularisation. And it is possible to construct non-collinear safe observables in the photonic case which leads to additional complications. We leave this for a future project.

Appendix A

New Subroutines in MadDipole

For the inclusion of MadDipole in the existing MadGraph code it was necessary to add new files with additional subroutines. In this appendix we are going to discuss the individual subroutines to explain what purpose they are for and how their principle algorithms work. We start with a short overview of the MadGraph code which enables us to distinguish the different levels at which insertions had to be made. After that we will then discuss the individual levels and their routines.

A.1 Structure of MadGraph

To understand at which point which routine has to added it is useful to first get the general idea about the structure of the original MadGraph program. The principle structure is illustrated in Fig.A.1. In this context the notions MadGraph and MadDipole can be used interchangeably. The original program is of course MadGraph whereas MadDipole acts as an add-on program. The way they work is however identical the only difference is that MadDipole provides additional files, the dipoles. The MadGraph program consists of several folders. The `MadGraphII` folder contains the actual code for the program. After compiling it also contains the executables. The aim of this program is to produce files which are able to calculate the squared matrixelement of a certain process the user has specified before running the code.
This is done with the help of the `Template` folder. First it is recommended to copy the whole content of this folder into a new one so that one always has a clean copy. We label this new folder with `MyProcDir` but the different name is the only difference between the two folders.
Within the `MyProcDir` folder the user specifies the process he wants to have calculated. This is done by editing and modifying the file `/Cards/proc_card.dat`. After specifying the process the MadDipole program is run from within this folder. The code contained in `MadGraphII` the produces several process dependent files and copies them into a subdirectory of the `MyProcDir` directory, which is labelled

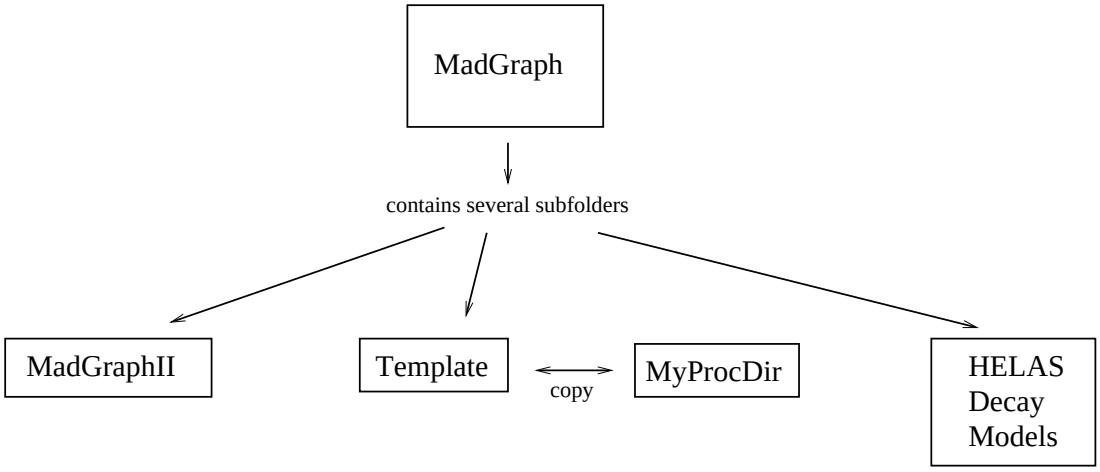


Figure A.1: Principle structure of MadGraph. The folder `MadGraphII` contains the program itself. It is recommended to copy the Template folder into a new folder, here called `MyProcDir`. The program is run from within that folder and the output code is copied into that folder.

with `/SubProcesses/P1_ab_cd`. The actual name of that subdirectory depends on the process. These files provided by the program contain a subroutine that calculates the squared matrixelement for the desired NLO process (`matrix.f`) as well as all the dipoles (the single dipoles are contained in the files `dipolxxx.f` where `xxx` stands for a number starting with `001`. In many cases one is interested in the sum of all dipoles which is more convenient to handle, so all the single dipoles are added in `dipolsum.f`. For the integrated dipoles the equivalent file to `dipolsum.f` is the file `intdipoles.f`. After these files are written they are copied into the `/SubProcesses/P1_ab_cd/` directory.
The interplay between the two parts of the code is shown in Fig.A.2

The idea is now that after running the MadGraph or MadDipole program the directory `MyProcDir` contains all the necessary files to form a self-consistent part of a program that is framework-independent. That means it can be plugged in into another program for a full NLO calculation.
From this point of view it is clear that we can split the following discussion into two separate parts. The first part deals with the question in what why the original MadGraph program needed to be modified and which parts had to be added to produce the code for the dipoles. The second question is then how do this files work and which additional process independent files were needed for the `MyProcDir` directory to form a self-consistent program as required.

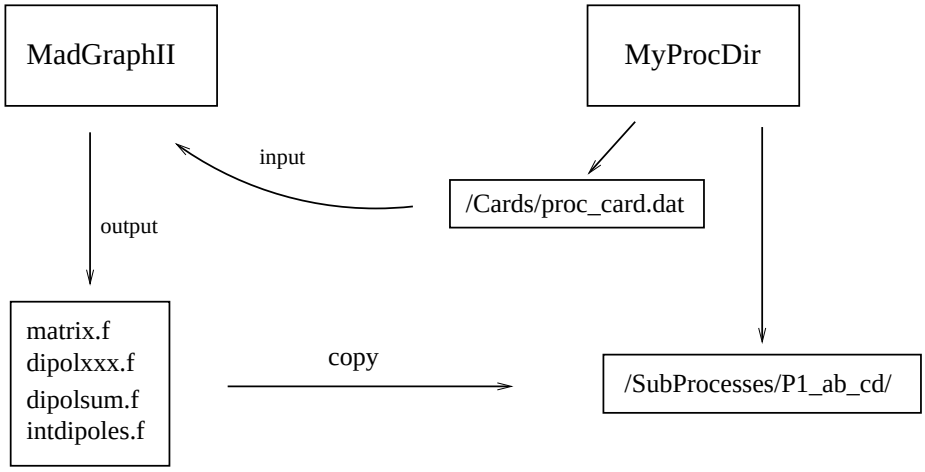


Figure A.2: Interplay between `MyProcDir` and `MadGraphII`. First a specific process is requested which leads to the production of the necessary process dependent files that are then copied into the subdirectory `/SubProcesses/P1_ab_cd/`, where `a` and `b` stand for the initial state particles and `c` and `d` stand for the final state particles.

A.2 Modification of main program

The output of the original MadGraph code is the file `matrix.f` that calculates the squared matrixelement for a given phase space point for a certain process.
If one is interested in the NLO real corrections not only the dipoles are needed but also the matrixelement for the $n + 1$ process. This is of course exactly given by the output of the original code. Therefore our modifications were done in such a way that instead of changing the original program a second program was added with the task to produce the dipoles. That means MadDipole actually performs two runs: The user specifies his NLO real correction process. Based on this the original MadGraph code produces the file `matrix.f`. This is followed by running a second program called `dipolterms`, the actual MadDipole program which produces all possible dipoles to all possible LO processes, the sum of all dipoles and the integrated dipoles.
Besides some modifications in several already existing files two new files were added to the `MadGraphII` directory. The program `Dipolterms` that is responsible for producing the dipoles is contained in a file called `maindipol.f`. Several other subroutines are needed to run the program, they are all contained in the file `dipolcolor.f`. In the following we will discuss the two files and the main properties of the subroutines they contain.

A.2.1 maindipol.f

Program Dipolterms

This is the main program for producing the dipoles. It contains only three different subroutines.

```
PROGRAM Dipolterms
⋮
call madgraph()
call write_dipoles()
call write_intdipoles()
⋮
END
```

The first routine is the main program for constructing the dipoles as described below. The second and the third routines are contained in `dipolcolor.f`. They create the files `dipolsum.f` and `intdipoles.f`. We will discuss them in more detail in the next section where we discuss the routines of `dipolcolor.f`

Subroutine madgraph

This subroutine is to a large extent identical with the original MadGraph program, hence the same name. Basically two things are done within this subroutine. Starting with the given NLO process it checks which combinations of emitter and spectator are possible, i.e. which dipoles exist. This information is stored in an object called

- `info_ijk(nr,2,5)` .

It returns a single integer number. The first entry `nr` is the current number of the dipole starting with 1. The second variable can be either 1 or 2. The first entry contains the number of the particle where the numbering of particles starts with 1 and the ordering is identical to the order the user has specified the process. It is also important to know whether the particle is a gluon or a quark. This information can be extracted if the second entry is chosen to be 2. In return the user gets either 0 or 1. If it is 0 then the particle is a gluon if it is 1 then the particle is a quark or an antiquark. At this point it is not possible to distinguish a quark from an antiquark. There are other subroutines which are able to do this. We will discuss this later.
The third entry specifies the role of the particle. Setting it to 1 addresses the emitter, 2 is the unresolved particle, 3 is the spectator, whereas 4 and 5 are the the combined particles $\widetilde{ij}$ and $\tilde{k}$.
We illustrate this with an example: If we want to know the number of the emitter in the first dipole, we would ask for `info_ijk(1,1,1)`. The question whether the spectator in the fifth dipole is a gluon or not can be answered by asking for

`info_ijk(5,2,3)` which will return a zero if it is indeed a gluon.

The rest of the subroutine is dedicated to the construction of the individual dipoles. The only difference compared to the original MadGraph program is the modified color structure of the dipoles that has to be taken into account. The subroutines responsible for the color calculation of the dipoles are in `dipolcolor.f` so we do not discuss them now. The result is then written in the files `dipolxxx.f` for each single dipole. This second part of the subroutine is constructed as a loop going through the list of existing dipoles stored in `info_ijk` and repeating the calculations of the dipole for each entry in `info_ijk`.

Subroutine square

A very similar subroutine with the same name already exists in the original program. The present version is indeed based on the original version including some modifications for the color calculation. The syntax for this subroutine is given by

- `square(flows,sflows,emitter,spectator)` .

Input variables are the integer numbers `emitter` and `spectator` and

- `integer flows(0:2*maxlines,0:maxfactors,0:24,0:maxflows)` .

This object is similar to the `color` object discussed below. Whereas `color` contains color information for a single amplitude, `flows` contains color information for all amplitudes.
The subroutine takes the color structure of an amplitude as the input and calculates the color factor of a matrix element squared out of it. The output of the calculation is then stored in `sflows` which is defined by

- `double precision sflows(maxflows,maxflows)` .

The parameters `maxlines`,`maxfactors` and `maxflows` are defined in the include file `params.inc`.
We describe the way and the systematics the color calculation is done in more detail in the next section when we discuss the individual routines dealing with the color structure and therefore omit a detailed discussion at this point.
As the subroutine contains several subroutines discussed below we give the general structure of the routine.

```
SUBROUTINE square(flows,sflows,emitter,spectator)
⋮
DO iflow=1,nflows
   call storecolor(color,iflow)
ENDDO
⋮
call insertdipol(color,emitter,spectator)
call doublecolor_tot(color,emitter,spectator)
⋮
DO iflow=1,nflows
  DO cflow=iflow,nflows
   call squarecolor_tot(color,color_tot,sqcolor,iflow,
                        cflow,max_ifactor,emitter,spectator)
   call dipolcolormatrix(sqcolor,color,iflow,cflow,
                                  emitter,spectator)
  ENDDO
ENDDO
⋮
END
```

A.2.2 dipolcolor.f

The file `dipolcolor.f` contains several subroutines. Most of them are needed to calculate the modified color structure.
In order to explain why these routines are needed we start with looking at a simple example and work out the necessary steps to calculate the color factor of a dipole. As it turns out this differs in practice from the color factor calculation of a normal amplitude. From there it will become clear that it was essential to introduce the additional routines.

We look at the process $e^+\ e^- \to u\ \bar{u}\ g$. Fig.A.3 shows one of the two amplitudes contributing to the total amplitude. We are only interested in the color factor. The conventional strategy for calculating the color factor is to write down the color structure of the amplitude, to sum over internal color lines, and then combine the result with the color structure of the complex conjugate amplitude. Summing over external color lines the leads to the complete color factor. For our example this is given by

$$\sum_{i,k=1..3,\ a=1..8} \boldsymbol{T}^a_{ik} \cdot \boldsymbol{T}^a_{ki} = Tr(\boldsymbol{T}^a \boldsymbol{T}^a) = \sum_a \frac{1}{2}\delta^{a,a} = C_{\rm F} N_{\rm c} = 4\ . \qquad \text{(A.1)}$$

The conclusion we want to draw here is that there is alway a prescription how to get the color structure of the complex conjugate amplitude when one has determined

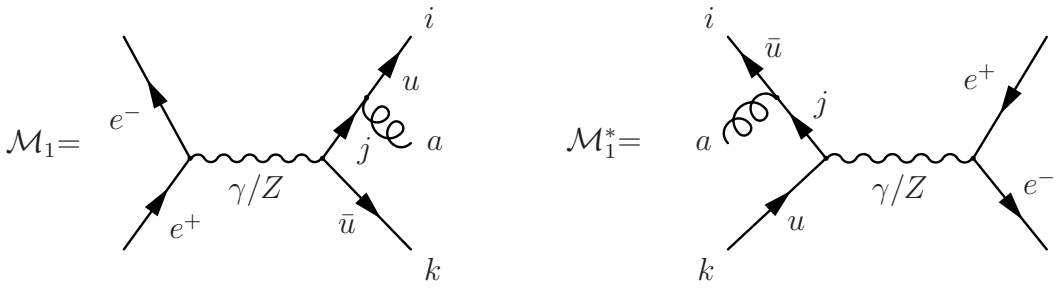


Figure A.3: The color structure for the amplitude on the left hand side is related to the color structure of the complex conjugate amplitude on the right hand side by switching the order of the fundamental indices i and k.

the color structure of the amplitude. In this example it is sufficient to switch the order of the fundamental indices i and j. Then one just needs to put the two pieces together and sum over the external indices.

The situation now changes when looking for the color structure of a dipole.
For this purpose it is useful to write an amplitude as a vector in color space. If we have n external particles with color indices $c_1 \cdots c_n$ we can express the amplitude as

$$\mathcal{M}_n^{c_1 \cdots c_n} = \langle c_1 \cdots c_n | 1 \cdots n \rangle_n \ . \tag{A.2}$$

This means that $|1 \cdots n\rangle_n$ is a vector in color space and the matrix element squared, summed over colors is then just given by

$$|\mathcal{M}_n|^2 = \ {}_n\langle 1 \cdots n | 1 \cdots n \rangle_n \ . \tag{A.3}$$

One can then define a color charge operator that acts on a vector in color space. If a parton i emits a gluon with color index a the action of the color charge operator on the color space is defined by

$$\langle c_1 \cdots c_i \cdots c_n | \boldsymbol{T}_i | b_1 \cdots b_i \cdots b_n \rangle = \delta_{c_1,b_1} \cdots \boldsymbol{T}^a_{c_i,b_i} \cdots \delta_{c_n,b_n} \ . \tag{A.4}$$

As seen in the previous chapters a dipole contains two color charge operators. A matrix element squared modified by two color charge operators can be written as

$$\langle c_1 \cdots c_i \cdots c_k \cdots c_n | \boldsymbol{T}_i \cdot \boldsymbol{T}_k | b_1 \cdots b_i \cdots b_k \cdots b_n \rangle = \delta_{c_1,b_1} \cdots \boldsymbol{T}^a_{c_i,b_i} \boldsymbol{T}^a_{c_k,b_k} \cdots \delta_{c_n,b_n} \ . \tag{A.5}$$

Coming back to our simple example we now want to calculate the color structure of the dipole where the quark is the emitter and the antiquark is the spectator. The underlying born process is $e^+ \ e^- \to u \ \bar{u}$. This situation is shown in Fig. A.4. As can be seen from Fig. A.4 and A.5 the additional color charge operators connect amplitude and the complex conjugate amplitude, i.e. there is a connection between

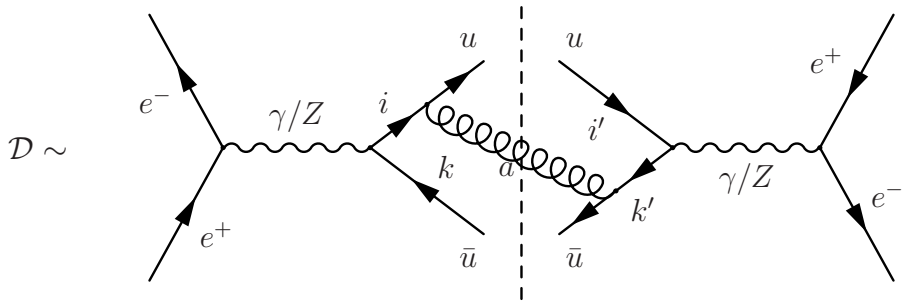


Figure A.4: Color structure of a dipole. The underlying born process is $e^+ e^- \to u \bar{u}$. Emitter and spectator and spectator are connect by an additional gluon. This requires different labelling of the color indices for the amplitude on the left hand side and the complex conjugate amplitude on the right hand side, in this case chosen to be i, k and i', k' respectively.

bra and ket in Eq. A.5. But this means that it is no longer possible to write down the color structure of the amplitude and then conclude the color structure of the complex conjugate amplitude. As Eq.A.5 indicates it is necessary to introduce a different labelling of the color indices of emitter and spectator for the bra and for the ket. All the other indices can be the same which avoids an overload of unnecessary Kronecker deltas.
The strategy that MadDipole uses is based on these considerations. First determine the color structure of the born process, then insert the additional color charge operators with introducing two new color indices for emitter and spectator for the complex conjugate amplitude. Out of the color structure of the born amplitude, the color structure of the complex conjugate amplitude is constructed with the difference that the color indices of emitter and spectator are replaced by the new ones. In a last step the pieces are then put together and the summation over the indices takes place.

Keeping this considerations in mind we will now discuss the individual routines contained in `dipolcolor.f`. Most of them contribute to the problem of dealing with the modified color structure.

Subroutine storecolor

The whole color information of the original MadGraph program is stored in object called `color`, defined as

- `integer color(0:2*maxlines,0:maxfactors,0:maxterms)` .

It is part of a common block and therefore available everywhere. The parameters `maxlines`, `maxfactors` and `maxterms` are defined in the include file `params.inc`.
As discussed above for the dipoles it is necessary to store the color information of amplitude and complex conjugate amplitude separately. So the existing color object

is no longer sufficient. This subroutine now copies the color information into a new enlarged object defined by

- `integer color_tot(2,200,0:2*maxlines,0:maxfactors,0:maxterms)` ,

which is also part of a common block to make it accessible throughout the whole program.
The routine itself is defined as

- `storecolor(color,iflow)` .

The procedure of copying is done individually for each amplitude which are numbered with the integer `iflow`. So both `color` and `iflow` are input variables. No output variable is needed because the information is written into a common block.

Subroutine getflowinfo1

This subroutine provides the information whether the external particle are incoming or outgoing and whether they are particles or anti particles. The information is stored in the object

- `integer flowinfo(max_particles,2)` .

All the input and output variables are part of a common block therefore the subroutine does not need any arguments. The first argument of `flowinfo` labels the number of the external particle. The second argument can be either 1 or 2. Entry 1 determines the state of the particle, i.e. it returns 0 if the particle is in the initial state or 1 if the particle is in the final state. The second entry distinguishes particles from antiparticles, it returns either 1 if it is a particle or -1 if it is an antiparticle. If the particle is a gluon or a non-qcd particle this value is set to 0.
This information is required to determine the correct color charge operator. As color operators are constructed in such a way that the fundamental indices, say i and k of $\boldsymbol{T}^a_{ik}$, are ordered in the opposite direction of the particle flow it is important to know if a particle is incoming or outgoing and if it is a particle or an antiparticle to determine the flow correctly.

Subroutine insertdipol

One of the main important steps in the calculation of the color factors is the insertion of the additional color operators according to A.5. This is done within the present subroutine

- `Insertdipol(color,emitter,spectator)` .

The properties of `color` are already discussed above and is an input variable for this subroutine as well as the integer numbers `emitter` and `spectator`.
The additional color charge operators are inserted in the new enlarged color object `color_tot` which has also already been discussed above. The systematics of relabelling of the color indices of emitter and spectator is as follows. Originally all color indices are just labelled with their particle number starting with one and going to the total number of external particles. The ordering depends on the order the process was specified. Internal lines are labelled with a negative integer number. For the relabelling of emitter and spectator we add `max_particle` to the originally given number. Per default this value is set to 127. The adjoint index which is labelled with a in A.5 and Fig. A.4 belongs to an internal particle, hence needing a negative integer number. We choose this number to be -99 with the reasonable assumption that there will never be more than 98 internal particles and therefore the value -99 is never assigned anywhere else.
If the emitter for instance is particle 4 and if this particle is an outgoing fermion then in our notation the additional color charge operator is given by $\boldsymbol{T}^{-99}_{131,4}$.

Subroutine doublecolor_tot

After inserting the color operators for the dipole the color structure for the complex conjugate amplitude is required. This is done by the subroutine `doublecolor_tot`,

- `doublecolor_tot(color,emitter,spectator)` .

It takes the original color structure of the amplitude out of which the color structure of the conjugate amplitude is constructed. The indices of emitter and spectator in the color structure of the conjugate amplitude are also relabelled at this point. This information is also stored in the enlarged color object `color_tot`. The first entry of `color_tot` distinguishes the color of the amplitude from the color of the conjugate amplitude. `color_tot(1,...)` is the color structure of the amplitude plus the additional color charge operators of the dipole and `color_tot(2,...)` denotes the color structure of the complex conjugate amplitude of course without additional operators.

Subroutine squarecolor_tot

So far we have constructed all possible color structures for each amplitude including the additional operators coming from the dipole and the color structures of the complex conjugate amplitude. The next step is now putting together all possible combinations of amplitudes with a complex conjugate amplitude,done by

- `squarecolor_tot(color,color_tot,sqcolor,j,i,`
  `max_ifactor,emitter,spectator)` .

The result is stored in the object

- `integer sqcolor(0:2*maxlines,0:maxfactors,0:maxterms)` ,

which already exists in the original code. This is the output variable of the subroutine. The input variables besides the already known variables `color`, `color_tot`, `emitter` and `spectator` are the integer numbers `i`, `j` and `max_ifactors(maxterms)`. `i` and `j` just denote which color structure of an amplitude is combined with which color structure of the conjugate amplitude.
`max_ifactors(maxterms)` is a technical parameter needed for the internal algorithm and is just the maximal numbers of color operators that occur for a specific amplitude.

Subroutine dipolcolormatrix

After having built all possible combinations the last thing that is missing to have the color matrix according to (3.1) which is written in the `matrix.f` file is to sum over all internal and external color indices. This is done in this subroutine,

- `dipolcolormatrix(sqcolor,color,i,j,emitter,spectator)` .

`sqcolor`, `i`, `j`, `emitter` and `spectator` are the input variables. In a first step the matrix `sqcolor` is copied into the old color matrix `color`. After that it is summed over all internal and external indices. What remains is a color matrix that contains real numbers and which is identical to the color matrix in (3.1). `color` is used as the output variable. The summation is performed for each possible combinations of amplitude and conjugate amplitude separately, therefore the summation indices `i` and `j` occur as for the construction of the squared color matrix `sqcolor` in the previous section. It is done using the already existing routines for summation. This is possible because at this stage there is no difference in the color structure compared to a LO process with one more particle.
Concerning the order of the summation one first has to sum over internal particles. Our starting point where we begin to modify the color structure was the already existing color structure of the underlying born process. There it has already been summed over internal particles so the only internal particle that is remaining is the additional 'particle' that is inserted by the dipole (the gluon with index a in A.4). So we start our summation by summing over this index followed by the summation over the relabelled indices of emitter and spectator and then we sum over the rest of the external particles.

Subroutine removelines

This subroutine is necessary at the point where the program is checking which dipoles exist and its syntax is given by

- `removelines(iline_copy,nr)` .

A dipole is symmetric under the interchange of emitter and unresolved particle. If the program has found a possible dipole for a certain combination of emitter, unresolved particle and spectator and then finds a combination where the roles of emitter and unresolved particles are interchanged with the same spectator, then the latter combination is identical to the previous one and is therefore removed from the list of all possibilities.
Originally information about the process is contained in

- `integer iline(-maxlines:maxlines)` ,

where `iline(0)` is the number of external particles. Each particle type is associated with a certain number given by an internal labelling system. This number is stored in `iline(1...n)` for the n external particles.
`iline_copy` is defined by

- `integer iline_copy(-maxlines:maxlines,maxdipoles)` .

Compared to `iline` it has one additional argument that allows us to store the process information for each single dipole in one object. It is input and output variable at the same time.
The second argument `nr` is an integer number that denotes the number of the current dipole.

Subroutine write_dipols

For most applications it is quite convenient to have all dipoles in one file. This subroutines creates the file `dipolsum.f` which calls all dipoles and sums the individual contribution. The content of this file will be discussed in the next section. This file also includes the cut on the α-parameter as a θ-function. No arguments are needed for this subroutine.

Subroutine write_intdipoles

This is the equivalent subroutine as `write_dipols()` but for the integrated dipoles. The subroutine creates a file `intdipoles.f` which calculates individual contributions for the integrated dipoles and their sum. We also discuss the content of this file in detail in the next section. There are also no arguments required.

Subroutine printdipolcolor

Although this subroutine is not needed for the actual color calculation it is a useful routine for debugging purposes. It is defined as

- `printdipolcolor(color)` ,

and it writes out the single entries of the `color` matrix in a specific way that allows the experienced user to check the color matrix at arbitrary intermediate points during the calculation.

A.3 Output files

The main output of the MadDipole program is basically a set of four files as indicated in Fig.A.2. The file `matrix.f` contains a subroutine to calculate the matrix element squared for the NLO process specified in `proc_card.dat`. This is of course produced by the original MadGraph program so we omit a discussion of this file here. The code for the calculation of the dipoles is contained in the files `dipolxxx.f`. `xxx` is a running number starting with `001`. The structure of these files are very similar to the structure of `matrix.f`.
The file `dipolsum.f` sums the contributions of the individual dipoles. It also includes the cut on the α-parameter. `intdipoles.f` contains the sum of all individual integrated dipoles. The structure of this file is similar to `dipolsum.f`

A.3.1 dipolxxx.f

The files `dipolxxx.f` calculate a single dipole contribution. The exact code is not the same for all dipoles but depends on the fact whether the splitting function contains only diagonal terms or also off-diagonal terms. All the files consist of a subroutine called `SDIPOLXXX` and a real function `DIPOLXXX`. The basic structure of the subroutine is given by

```
SUBROUTINE SDIPOL001(P1,ANS)
⋮
c momentum remapping
IF(INTDIPOLE .EQV. .FALSE.) THEN
  CALL TRANSFORM(...)
ELSE
  CALL TRANSFORMINT(...)
ENDIF

c sum over hel. combinations and evaluate matrix element
c squared for single combination.
DO IHEL=1,NCOMB
 T=DIPOL001(...)
  ANS=ANS+T
ENDDO

c in case of off-diagonal terms a second summation is required
DO I=1,NCOMB
  DO L=1,NCOMB
  IF(NHEL(1,I).EQ.NHEL(1,L).AND.NHEL(2,I).EQ.NHEL(2,L).AND.
  NHEL(3,I).EQ.-NHEL( 3,L).AND.NHEL(4,I).EQ.NHEL(4,L)) THEN
    T=T+DREAL(STOREJAMP(I,1,K)*STOREJAMP(L,2,K))
  ENDDO
ENDDO
⋮
END
```

A major difference to the code in `matrix.f` is that the code has to remap the momenta of an $n+1$-particle process to n new momenta. This remapping is different for the dipoles and for the integrated dipoles. For the first case this is done in the subroutine `transform`, the latter is treated in `transformint`. We will discuss these subroutines in the next section.
The next step is summing over all helicity combinations and evaluating the dipole for each single helicity combination. This is done within the real function `DIPOLXXX`. In case that there are also off-diagonal terms contributing there is another summation over all helicities and the appropriate off-diagonal terms are added the the final result. In this example the emitter is given by particle with the number three because it is required that its helicity is different for amplitude and complex conjugate amplitude whereas for all other particles it is required to be the same. The input variable for this subroutine is the $n+1$ phase space point `P1(0:3,nexternal` where `nexternal` is the number of the external particles. The output variable is the matrix element squared `ANS`.

The exact structure of the real function DIPOLXXX also depends on the fact if there are off-diagonal terms or not. It is given by

```
REAL*8 FUNCTION DIPOL001(P,NHEL,IC,P1)
c if there are off-diagonal the function is called by
REAL*8 FUNCTION DIPOL001(P,NHEL,IC,P1,IHEL,STOREJAMP)
⋮
c here the color matrix is printed and the color-ordered amplitudes
c are calculated using HELAS routines. The result is an array
called JAMP.
JAMP(1)= ...
⋮
CALL DIPOLESUB(...,SUB,SUBV)
IF (INTDIPOLE .EQV. .TRUE.) THEN
  SUB=1d0
  SUBV=(0d0,0d0)
ENDIF

c now combining amplitudes and color matrix as in 3.1
DO I=1,NCOLOR
  DO J=1,NCOLOR
    ZTEMP=ZTEMP+CF(J,I)*JAMP(J)
  ENDDO
  DIPOL001=DIPOL001+ZTEMP*DCONJG(JAMP(I))/DENOM(I)*SUB
c if there are off-diag. terms there are additional contributions
 STOREJAMP(IHEL,1,I)=ZTEMP
 STOREJAMP(IHEL,2,I)=DCONJG(JAMP(I))/DENOM(I)*SUBV
ENDDO
END
```

The input variables of this function are the phase space point P(0:3,nexternal) containing the remapped momenta, the helicities of the external particles NHEL(nexternal) which are integer numbers. The integer variable IC(nexternal) is only needed if a particle needs to be flipped. As long as there is at least one incoming particle this variable is not needed and set to unity. Another input variable is the phase space point with the original momenta P1(0:3,nexternal).
In the case where there are also off-diagonal contributions two additional input variables are needed, namely the number of the current helicity configuration IHEL which is an integer number and the real array STOREJAMP(NCOMB,2,NCOLOR) where the color ordered amplitudes are stored. This storage is done for each helicity combination (NCOMB) and for each single color structure (NCOLOR). The argument in the

middle distinguishes the amplitude multiplied with the splitting function (1) from the complex conjugate amplitude (2).
Within this function the color matrix of 3.1 is printed and the color ordered amplitudes `JAMP()` are calculated. This is done using the HELAS routines [85]. Besides the fact that the color matrix is modified due to the additional color charge operators of the dipole this is identical to a common matrix element.
In the subroutine `dipolesub` the splitting function that is part of the dipole is calculated. It is contained in the file `dipolesub.f` which we will discuss later. The output variables are called `SUB` and `SUBV` which give the diagonal terms as a real number and the off-diagonal terms as a complex number respectively. When we are interested in the integrated dipoles this is not needed therefore the diagonal terms are set to one and the off-diagonal terms are set to zero.
Finally the amplitudes are combined with the color matrix to build the full matrix element squared multiplied by the diagonal part of the splitting function `SUB`. If there are also off-diagonal terms the amplitudes and their complex comjugates are stores in `STOREJAMP` where the complex conjugate amplitude is multiplied with the off-diagonal terms of the splitting function.

A.3.2 dipolsum.f

Subroutine dipolsum

This is the first of two subroutines contained in this file and it is called by

- `DIPOLSUM(P,SUBTRACT)` .

The input variable of this subroutine is the phase space point `P(0:3,nexternal)` and the output is the sum of all dipole contributions, a real number labelled as `subtract`. The cut on the α-parameter in form of θ-functions as shown in Eq. (3.6) is contained in this function. The θ-functions are realized in the form of if-statements. The relevant kinematical variable which parametrises the infrared divergence (e.g. $y_{ij,k}$ for a final-final dipole) is calculated for each dipole before checking the if-statement. The values for the individual α-parameters can be specified in the file `dipole.inc`.
The principle structure of this subroutine is given by

```
SUBROUTINE DIPOLSUM(P,SUBTRACT)
⋮
IF (alpha_ii .gt.vtilde_i) THEN
  CALL SDIPOL001(P,ANS( 1))
  SUBTRACT = SUBTRACT - ANS( 1)
ENDIF
⋮
IF (alpha_ff .gt.y_ijk) THEN
  CALL SDIPOL024(P,ANS( 5))
  SUBTRACT = SUBTRACT - ANS( 5)
ENDIF
⋮
END
```

Subroutine dipolsumfinite

As explained already in chapter three, MadDipole calculates the dipoles in cases where the singularity is in principle regulated by the masses of the particles and therefore a subtraction scheme not necessarily needed. This occurs for the splitting of a gluon into two massive quarks. These dipoles are written in this subroutine whose arguments and structure is exactly identical to `dipolsum`. Neglecting this subroutine will still lead to a finite result. However its use may be desirable to ensure numerical stability In the case where no such dipoles occur this routine is just given by an empty body.

A.3.3 intdipoles.f

This subroutine is called via

- `INTDIPOLES(P,Z,EPS,EPSSQ,FINITE)` .

The input variables are the phase space point `P(0:3,nexternal)` and the momentum fraction `Z` for intial states. The output variables are the coefficient of the $\frac{1}{\epsilon}$-term,`EPS`, the coefficient of the $\frac{1}{\epsilon^2}$-term and the five dimensional array for the finite terms, `FINITE`.
The main building block of this routine is given by

```
SUBROUTINE INTDIPOLES(P,Z,EPS,EPSSQ,FINITE)
⋮
CALL SDIPOL001(P,ANS( 1))
CALL EPSII( 1, 1,sikzone,esq,e)
  epssq=epssq-esq*ANS( 1)
  eps=eps-e*ANS( 1)
CALL FINITEII(sik,sikzone,z, 1, 1,fi,fi2)
  finite(1)=finite(1)-fi(1)*ANS( 1)
  finite(2)=finite(2)-fi(2)*ANS( 1)-fi2(2)*ANS( 1)
  finite(3)=finite(3)+fi(3)*ANS( 1)
```

The actual form is of course strongly dependent on the actual process but the basic strategy is the same: First the integrated dipole without the splitting function is calculated. In the second step a function is called that calculates the coefficients of the divergent terms. We describe this function in the next section. In a third step a function is called that calculates the finite parts. This routine is also described in the next section.
The entries od this array of finite terms is described in section 4.5. The example given here refers to massless particles that is why we do not have a contribution to the fourth and the fifth entry.

A.4 Additional files

The files and their subroutines discussed in the previous section contain additional subroutines whose discussion has been postponed so far. All additional files and subroutines which are necessary to embed the output files in a full program can be found in the `MyProcDir/SubProcesses` directory.

A.4.1 transform.f

Subroutine transform

The subroutine `transform` is responsible for the remapping of the momenta when going from $n+1$ particles of the full matrix element to the n particles of the dipole. Its arguments are

- `transform(p,q,ip,jp,kp,ijp,kpp,mass_i,mass_j,mass_k)` .

The input variables are the phase space point `p(0:3,nexternal)`, the numbers of the emitter, unresolved particle, spectator, combined particle of emitter/unresolved and the remapped spectator, labelled with `ip`, `jp`, `kp`, `ijp` and `kpp`. Further input variables are the masses of emitter, unresolved particle and spectator, given by

`mass_i`, `mass_j` and `mass_k`. The output is then given by the new phase space point `q(0:3,nexternal)`. Note that the dimensions of `p` and `q` are the same although the dipole has one particle less. The last entry of `q` is filled with zeros. Internally the subroutine distinguishes between massless and massive case.

real*8 function lambda_tr

For the transformation of the momenta in the massive case the triangular function is needed, defined as

$$\lambda = x^2 + y^2 + z^2 - 2xy - 2xz - 2yz \; . \tag{A.6}$$

The function is called by

- `lambda_tr(x,y,z)` ,

returning the value λ defined in (A.6), where the arguments are real numbers.

real*8 function dot

This function returns the scalar product of 4-vectors, defined as

$$p_1 \cdot p_2 = p_1^0 p_2^0 - p_1^1 p_2^1 - p_1^2 p_2^2 - p_1^3 p_2^3 \; . \tag{A.7}$$

It is called with the two arguments p_1 and p_2,

- `dot(p1,p2)` ,

where the momenta of the arguments must be given as `p(0:3)`.

A.4.2 transformint.f

Subroutine transformint

When evaluating the integrated dipoles no remapping of the momenta is needed because the integrated dipoles are called with a phase space point coming from a n particle phase space. So in principle this routine is not necessarily required but it serves more technical reasons. The `dipolxxx.f` files are used for the unintegrated dipoles as well as for the integrated dipoles. To be able to maintain the structure of the files we call this dummy routine with the same arguments as the `transform` routine, namely

- `transformint(p,q,ip,jp,kp,ijp,kpp,mass_i,mass_j,mass_k)` .

However it only copies the phase space point `p` into the new phase space point `q`.

A.4.3 dipolesub.f

Subroutine dipolesub

The subroutine `dipolesub` provides the splitting function that is multiplied with the color modified born-type process. The argument list of this subroutine is rather long, given by

- `dipolesub(p,ip,jp,kp,ijp,kkp,ig,jg,kg,ijg,kkg,mass_i,mass_j, mass_k,w1,w2,sub,subv)` .

The input variables are the phase space point `p(0:3,nexternal)` and as for the `transform` routine the numbers of emitter, unresolved particle, spectator, the number of the combination of emitter and unresolved particle and the new spectator, denoted by `ip`, `jp`, `kp`, `ijp`, `kkp`. Furthermore there is also information about the type of the particles needed, given by the integer number `ig`, `jg`, `kg`, `ijg` and `kkg`. They are either zero if the particle is a gluon or they are one if the particle is a quark or an antiquark. The masses are also needed as input variables, given by `mass_i`, `mass_j` and `mass_k`.
For the off-diagonal terms also the polarisation vector of the emitter and the polarisation vector with the opposite helicity is needed as can be seen from the discussion in Eq.(3.5). These are the input variables denoted by `w1` and `w2`. They are evaluated using HELAS routines and therefore they are given by an 18-dimensional complex array.
The output is given by the real number `SUB` for the diagonal terms and the complex number `SUBV` for the off-diagonal terms.
Also this subroutine internally distinguishes between massless and massive case using two separate calculations.

complex*16 function dotrc

For the calculation of the splitting function in `dipolesub` the scalar product of a momentum with the polarisation vector is needed. As the polarisation vector is given as a complex array, the function for the scalar product described above cannot be used. This function here is called by

- `dotrc(p,v)` ,

where the first argument `p(0:3)` is a real array but the second argument `v(0:3)` has to be a complex array.

complex*16 function dotcc

This is the same function as the previous one but now also the first argument has to be a complex array.

A.4.4 epsterms.f

The file `epsterms.f` contains subroutines to calculate the coefficients of the divergent terms as defined in chapter 4.2 when integrating over the phase space of the unresolved particle. The four different cases final-final, final-initial, initial-final and initial-initial are treated separately.

Subroutine epsff

This subroutine calculates the divergent terms for the final-final case. It is called by

- `epsff(mi,mk,sik,id,id1,epssq,eps)` .

As input variables this routine needs the masses of emitter and spectator, `mi` and `mk` and the information about the particle type of the emitter and the combined particle made of emitter and unresolved particle. `id` denotes the type of the combined particle and it is zero if the particle is a gluon or one if it is a quark or an anti quark. `id1` is the same for the emitter. The variable `sik` is only needed for the massive case and is given by the kinematical invariant that goes into the definition of the rescaled masses μ. For the final-final case this is just $\sqrt{Q^2}$ with $Q^\mu = \tilde{p}_{ij}^\mu + \tilde{p}_k^\mu$. The output variables are then the coefficients of the $1/\epsilon^2$ terms (`epssq`) and $1/\epsilon$ terms (`eps`).

Subroutine epsfi

Here the coefficients of the divergent terms are calculated for the final-initial case. The subroutine is defined as

- `epsfi(id,id1,sikzone,mi,epssq,eps)` .

As we require initial state particles to be massless only the mass of the emitter `mi` is required. The meaning of `id` and `id1` is identical to the final-final case. In this case `sik` is defined as $2\tilde{p}_{ij}p_a/z$.

Subroutine epsif

For the initial-final case the corresponding subroutine is given by

- `epsif(id,id1,sikzone,mk,epssq,eps)` .

Again the meaning of the variables is identical to the previous cases. Here only the mass of the spectator `mk` is required as the emitter has to be massless. The variable `sik` for the initial-final case is defined as $2\tilde{p}_j p_a/z$.

Subroutine epsii

The last remaining case is the initial case. As there must not occur any masses the subroutine reads as

- `epsii(id1,id2,sikzone,epssq,eps)` .

For the initial-initial case the variable `sik` is given by $2p_a p_b/z$. All the other arguments are the same as before.

A.4.5 finiteterms.f

This file contains the necessary subroutines for calculating the finite terms. As described in chapter four we offer two different possibilities concerning the expansion of the overall prefactors. For reasons of clarity each possibility is treated in a separate function. For the divergent terms all expressions are very short so this was not needed. If the user chooses the method where the prefactor is expanded the relevant routines are `finiteff`, `finitefi`, `finiteif` and `finiteii`. For the other case the corresponding routines are `finiteffI1`, `finitefiI1`, `finiteifI1` and `finiteiiI1`. Note however that in any case first the former subroutines are called and there it is checked if it is necessary to switch to the other routines. The reason is that at the time where `intdipoles.f` is written it is not known which method is chosen. The arguments of the two types and the way they are structured is the same such that we restrict ourselves to discussing only the former ones. Both types of routines take into account that the finite terms depend on the choice of the regularisation scheme and add the appropriate terms if necessary.
These routines also include the dependence on the cut parameter α. The additional contributions are added automatic*ally to the other finite terms which give the correct result for* $\alpha = 1$. *The* current value for the α parameter is known due to the inclusion of `dipole.inc` which we will discuss below.

Subroutine finiteff

We start with the appropriate routine for the final-final case. It is given by

- `finiteff(mi,mk,sik,id,id1,finite)` .

The input variables `mi`, `mk`, `sik`, `id` and `id1` are identical to the ones discussed for the divergent terms of this case. As the finite terms for the final-final case cannot contain any distribution it is sufficient to return a single real number `finite`.

Subroutine finitefi

The situation is different if at least one particle is in the initial state. For the final-initial case the subroutine is given by

- `finitefi(mi,sik,sikzone,z,id,id1,finite,fi2)` .

The input variables `mi`, `sik`, `id` and `id1` remain unchanged. Again a mass can only occur for the emitter. There are however important differences to what we have discussed so far. First there is an additional variable `z` which denotes the momentum fraction that remains for the spectator after rescaling the momentum. The finite terms now contain not only terms which are regular functions in z but also distributions. This means that objects have to be calculated for a given value of z and at $z = 1$. The variable `sik` implicitly depends on z by the way it is defined. Its value is therefore different at $z = 1$. This is taken into account by introducing a new variable `sikzone` which exactly corresponds to `sik` at the point where z is equal to one. The output variable `finite` is now a five-dimensional array according to the discussion in chapter four.
In some massive cases products of a regular function and a plus-distribution occur. These cases are a bit tricky to handle because for the plus-distribution term the endpoint has to be subtracted while keeping the regular terms fixed. These terms are treated separately in a five dimensional array `fi2`. They are added to the other finite terms in an appropriate way in the `intdipoles` subroutine.

Subroutine finiteif

The initial-final case is very similar to the final-initial case. Replacing the emitter mass by the spectator mass the subroutine is defined as

- `finiteif(mk,sik,sikzone,z,id,id1,finite,fi2)` .

In this case the momentum fraction `z` refers to the rescaled emitter instead of the spectator but otherwise the definitions and meaning of the arguments are identical to the former final-initial case.

Subroutine finiteii

In the initial-initial case we can also neglect the mass of the spectator hence the subroutine is

- `finiteii(sik,sikzone,z,id1,id2,finite,fi2)` .

Also here the momentum fraction `z` refers to the rescaling of the emitter.

A.4.6 dipole.inc

This file is designed to be an include file to make important parameters accessible wherever they are needed. Besides some technical variables that are stored in a common block and contained in the file it contains besides

- the renormalization scale,
  ```
  real*8 mu
  parameter (mu=91.188d0)
  ```
- the factorization scale,
  ```
  real*8 muf
  parameter (muf=91.188d0)
  ```
- the number of different flavors,
  ```
  integer Nf
  parameter (Nf=1)
  ```

also the α parameters for cutting the phase space. They are defined as

- ```
  real*8 alpha_ff,alpha_fi,alpha_if,alpha_ii
  parameter (alpha_ff=1d0)
  parameter (alpha_fi=1d0)
  parameter (alpha_if=1d0)
  parameter (alpha_ii=1d0).
  ```

Furthermore the user can specify which regularisation scheme he wants to use. It can be adjusted by setting

- ```
  character*2 scheme
  parameter (scheme='HV')
  ```

for the 't Hooft-Veltman scheme or using four dimensional helicity scheme by replacing `'HV'` by `'DR'`.
Also the method of treating the common prefactors in the integrated dipoles can be specified here. For this reason the variable

- ```
  logical I1
  parameter (I1=.FALSE.)
  ```

has been created. Setting `I1` to `.FALSE.` means that the whole prefactor is expanded whereas setting `I1` to `.TRUE.` means factorizing out an overall prefactor as defined in chapter four. This coincides with using the routines with the ending `...I1` for the finite terms.
  ```

Acknowledgments

First of all I am very grateful to my supervisor Prof. Dr. Thomas Gehrmann for giving me the chance and the possibility to do my PhD in his group. Besides the very interesting topic he gave to me he was always encouraging me to be not too restricted to my research field but to broaden my view which has lead to interesting projects [121, 122]. He always took the time for discussions and advises and answered all my questions no matter how stupid they were and he was a perfect guide for my struggle with this thesis. I am very glad that I had the luck of having such an excellent supervisor.

I also want to thank the other members of the committee, Prof. Dr. Ben Moore and Prof. Dr. Daniel Wyler for their time and their efforts concerning this thesis.

Furthermore I thank my office mates Dr. Daniel Maitre, Pedro Schwaller and Nurhana Tajuddin for the nice and good atmosphere. It was a great pleasure to be part of this office 36k36.

A special thanks goes to the computer administrators, in particular Doug Potter and Dr. Roland Bernet who took good care of all my problems with hard- and software.

I fear that I cannot mention everybody who deserves to be mentioned. However I do not want to forget mentioning Dr. Daniel Maitre for introducing me to the Institute and for revealing me the secrets of Doppelkopf.
I also do not want to forget to thank in particular Dr. Tobias Goerdt, Tobias Motz, Katja and Pedro Schwaller and all the members of the Institute for various unforgetable 'excursions' that I would not want to miss.

But most of all I want to thank Anuschka Gabriel for her love, understanding and her never ending support throughout all the years and for giving me the stability that was necessary for this work.

Bibliography

[1] G. 't Hooft and M. J. G. Veltman, "Regularization and Renormalization of Gauge Fields," *Nucl. Phys.* **B44** (1972) 189–213.

[2] F. Bloch and A. Nordsieck, "Note on the Radiation Field of the electron," *Phys. Rev.* **52** (1937) 54–59.

[3] T. Kinoshita, "Mass singularities of Feynman amplitudes," *J. Math. Phys.* **3** (1962) 650–677.

[4] T. D. Lee and M. Nauenberg, "Degenerate Systems and Mass Singularities," *Phys. Rev.* **133** (1964) B1549–B1562.

[5] W. T. Giele and E. W. N. Glover, "Higher order corrections to jet cross-sections in e^+ e^- annihilation," *Phys. Rev.* **D46** (1992) 1980–2010.

[6] Z. Kunszt and D. E. Soper, "Calculation of jet cross-sections in hadron collisions at order α_s^3," *Phys. Rev.* **D46** (1992) 192–221.

[7] S. Frixione, Z. Kunszt, and A. Signer, "Three-jet cross sections to next-to-leading order," *Nucl. Phys.* **B467** (1996) 399–442, `hep-ph/9512328`.

[8] D. A. Kosower, "Antenna factorization of gauge-theory amplitudes," *Phys. Rev.* **D57** (1998) 5410–5416, `hep-ph/9710213`.

[9] A. Daleo, T. Gehrmann, and D. Maître, "Antenna subtraction with hadronic initial states," *JHEP* **04** (2007) 016, `hep-ph/0612257`.

[10] A. Gehrmann-De Ridder, T. Gehrmann, and E. W. N. Glover, "Antenna subtraction at NNLO," *JHEP* **09** (2005) 056, `hep-ph/0505111`.

[11] J. M. Campbell, M. A. Cullen, and E. W. N. Glover, "Four jet event shapes in electron positron annihilation," *Eur. Phys. J.* **C9** (1999) 245–265, `hep-ph/9809429`.

[12] S. Catani and M. H. Seymour, "A general algorithm for calculating jet cross sections in NLO QCD," *Nucl. Phys.* **B485** (1997) 291–419, `hep-ph/9605323`.

[13] S. Catani, S. Dittmaier, M. H. Seymour, and Z. Trocsanyi, "The dipole formalism for next-to-leading order QCD calculations with massive partons," *Nucl. Phys.* **B627** (2002) 189–265, hep-ph/0201036.

[14] H. Fritzsch, M. Gell-Mann, and H. Leutwyler, "Advantages of the Color Octet Gluon Picture," *Phys. Lett.* **B47** (1973) 365–368.

[15] D. J. Gross and F. Wilczek, "Asymptotically Free Gauge Theories. 1," *Phys. Rev.* **D8** (1973) 3633–3652.

[16] T. Stelzer and W. F. Long, "Automatic generation of tree level helicity amplitudes," *Comput. Phys. Commun.* **81** (1994) 357–371, hep-ph/9401258.

[17] F. Maltoni and T. Stelzer, "MadEvent: Automatic event generation with MadGraph," *JHEP* **02** (2003) 027, hep-ph/0208156.

[18] J. Alwall *et al.*, "MadGraph/MadEvent v4: The New Web Generation," *JHEP* **09** (2007) 028, 0706.2334.

[19] **CompHEP** Collaboration, E. Boos *et al.*, "CompHEP 4.4: Automatic computations from Lagrangians to events," *Nucl. Instrum. Meth.* **A534** (2004) 250–259, hep-ph/0403113.

[20] A. Pukhov, "CalcHEP 3.2: MSSM, structure functions, event generation, batchs, and generation of matrix elements for other packages," hep-ph/0412191.

[21] T. Gleisberg *et al.*, "SHERPA 1.α, a proof-of-concept version," *JHEP* **02** (2004) 056, hep-ph/0311263.

[22] W. Kilian, "WHIZARD 1.0: A generic Monte-Carlo integration and event generation package for multi-particle processes. Manual,". LC-TOOL-2001-039.

[23] M. L. Mangano, M. Moretti, F. Piccinini, R. Pittau, and A. D. Polosa, "ALPGEN, a generator for hard multiparton processes in hadronic collisions," *JHEP* **07** (2003) 001, hep-ph/0206293.

[24] A. Kanaki and C. G. Papadopoulos, "HELAC: A package to compute electroweak helicity amplitudes," *Comput. Phys. Commun.* **132** (2000) 306–315, hep-ph/0002082.

[25] J. M. Campbell and R. K. Ellis, "An update on vector boson pair production at hadron colliders," *Phys. Rev.* **D60** (1999) 113006, hep-ph/9905386.

[26] J. Campbell and R. K. Ellis, "Next-to-leading order corrections to W + 2jet and Z + 2jet production at hadron colliders," *Phys. Rev.* **D65** (2002) 113007, hep-ph/0202176.

[27] Z. Nagy, "Next-to-leading order calculation of three-jet observables in hadron hadron collision," *Phys. Rev.* **D68** (2003) 094002, hep-ph/0307268.

[28] S. Frixione and B. R. Webber, "Matching NLO QCD computations and parton shower simulations," *JHEP* **06** (2002) 029, hep-ph/0204244.

[29] S. Frixione and B. R. Webber, "The MC@NLO 3.3 event generator," hep-ph/0612272.

[30] P. Nason, "A new method for combining NLO QCD with shower Monte Carlo algorithms," *JHEP* **11** (2004) 040, hep-ph/0409146.

[31] P. Nason and G. Ridolfi, "A positive-weight next-to-leading-order Monte Carlo for Z pair hadroproduction," *JHEP* **08** (2006) 077, hep-ph/0606275.

[32] O. Latunde-Dada, S. Gieseke, and B. Webber, "A positive-weight next-to-leading-order Monte Carlo for e^+e^- annihilation to hadrons," *JHEP* **02** (2007) 051, hep-ph/0612281.

[33] S. Frixione, P. Nason, and G. Ridolfi, "A Positive-Weight Next-to-Leading-Order Monte Carlo for Heavy Flavour Hadroproduction," *JHEP* **09** (2007) 126, 0707.3088.

[34] S. Alioli, P. Nason, C. Oleari, and E. Re, "NLO vector-boson production matched with shower in POWHEG," *JHEP* **07** (2008) 060, 0805.4802.

[35] K. Hamilton, P. Richardson, and J. Tully, "A Positive-Weight Next-to-Leading Order Monte Carlo Simulation of Drell-Yan Vector Boson Production," 0806.0290.

[36] D. B. Melrose, "Reduction of Feynman diagrams," *Nuovo Cim.* **40** (1965) 181–213.

[37] Z. Bern, L. J. Dixon, and D. A. Kosower, "Dimensionally regulated pentagon integrals," *Nucl. Phys.* **B412** (1994) 751–816, hep-ph/9306240.

[38] Z. Bern, L. J. Dixon, D. C. Dunbar, and D. A. Kosower, "One loop N point gauge theory amplitudes, unitarity and collinear limits," *Nucl. Phys.* **B425** (1994) 217–260, hep-ph/9403226.

[39] Z. Bern and A. G. Morgan, "Massive Loop Amplitudes from Unitarity," *Nucl. Phys.* **B467** (1996) 479–509, hep-ph/9511336.

[40] Z. Bern, L. J. Dixon, and D. A. Kosower, "Bootstrapping multi-parton loop amplitudes in QCD," *Phys. Rev.* **D73** (2006) 065013, hep-ph/0507005.

[41] C. F. Berger, Z. Bern, L. J. Dixon, D. Forde, and D. A. Kosower, "Bootstrapping one-loop QCD amplitudes with general helicities," *Phys. Rev.* **D74** (2006) 036009, hep-ph/0604195.

[42] D. Forde, "Direct extraction of one-loop integral coefficients," *Phys. Rev.* **D75** (2007) 125019, 0704.1835.

[43] A. Denner, S. Dittmaier, M. Roth, and L. H. Wieders, "Electroweak corrections to charged-current $e^+ e^- \to 4$ fermion processes: Technical details and further results," *Nucl. Phys.* **B724** (2005) 247–294, hep-ph/0505042.

[44] A. Denner and S. Dittmaier, "Reduction schemes for one-loop tensor integrals," *Nucl. Phys.* **B734** (2006) 62–115, hep-ph/0509141.

[45] T. Binoth, J. P. Guillet, G. Heinrich, E. Pilon, and C. Schubert, "An algebraic / numerical formalism for one-loop multi-leg amplitudes," *JHEP* **10** (2005) 015, hep-ph/0504267.

[46] T. Binoth, J. P. Guillet, and G. Heinrich, "Algebraic evaluation of rational polynomials in one-loop amplitudes," *JHEP* **02** (2007) 013, hep-ph/0609054.

[47] R. Britto, B. Feng, and P. Mastrolia, "The cut-constructible part of QCD amplitudes," *Phys. Rev.* **D73** (2006) 105004, hep-ph/0602178.

[48] P. Mastrolia, "On triple-cut of scattering amplitudes," *Phys. Lett.* **B644** (2007) 272–283, hep-th/0611091.

[49] R. Britto and B. Feng, "Integral Coefficients for One-Loop Amplitudes," *JHEP* **02** (2008) 095, 0711.4284.

[50] R. Britto, B. Feng, and P. Mastrolia, "Closed-Form Decomposition of One-Loop Massive Amplitudes," 0803.1989.

[51] C. Anastasiou, R. Britto, B. Feng, Z. Kunszt, and P. Mastrolia, "Unitarity cuts and reduction to master integrals in d dimensions for one-loop amplitudes," *JHEP* **03** (2007) 111, hep-ph/0612277.

[52] R. K. Ellis, W. T. Giele, and Z. Kunszt, "A Numerical Unitarity Formalism for Evaluating One-Loop Amplitudes," *JHEP* **03** (2008) 003, 0708.2398.

[53] W. T. Giele, Z. Kunszt, and K. Melnikov, "Full one-loop amplitudes from tree amplitudes," *JHEP* **04** (2008) 049, 0801.2237.

[54] R. K. Ellis, W. T. Giele, Z. Kunszt, and K. Melnikov, "Masses, fermions and generalized D-dimensional unitarity," 0806.3467.

[55] G. Ossola, C. G. Papadopoulos, and R. Pittau, "Reducing full one-loop amplitudes to scalar integrals at the integrand level," *Nucl. Phys.* **B763** (2007) 147–169, hep-ph/0609007.

[56] G. Ossola, C. G. Papadopoulos, and R. Pittau, "On the Rational Terms of the one-loop amplitudes," *JHEP* **05** (2008) 004, 0802.1876.

[57] P. Mastrolia, G. Ossola, C. G. Papadopoulos, and R. Pittau, "Optimizing the Reduction of One-Loop Amplitudes," *JHEP* **06** (2008) 030, 0803.3964.

[58] G. Ossola, C. G. Papadopoulos, and R. Pittau, "CutTools: a program implementing the OPP reduction method to compute one-loop amplitudes," *JHEP* **03** (2008) 042, 0711.3596.

[59] C. F. Berger *et al.*, "An Automated Implementation of On-Shell Methods for One- Loop Amplitudes," 0803.4180.

[60] W. T. Giele and G. Zanderighi, "On the Numerical Evaluation of One-Loop Amplitudes: the Gluonic Case," 0805.2152.

[61] T. Binoth *et al.*, "Precise predictions for LHC using a GOLEM," 0807.0605.

[62] A. Bredenstein, A. Denner, S. Dittmaier, and S. Pozzorini, "NLO QCD corrections to pp → t anti-t b anti-b + X at the LHC," 0905.0110.

[63] J. M. Campbell, R. K. Ellis, and G. Zanderighi, "Next-to-leading order Higgs + 2 jet production via gluon fusion," *JHEP* **10** (2006) 028, hep-ph/0608194.

[64] R. Keith Ellis, K. Melnikov, and G. Zanderighi, "W+3 jet production at the Tevatron," 0906.1445.

[65] R. K. Ellis, K. Melnikov, and G. Zanderighi, "Generalized unitarity at work: first NLO QCD results for hadronic W^+ 3jet production," *JHEP* **04** (2009) 077, 0901.4101.

[66] R. K. Ellis, W. T. Giele, Z. Kunszt, K. Melnikov, and G. Zanderighi, "One-loop amplitudes for W^+ 3 jet production in hadron collisions," *JHEP* **01** (2009) 012, 0810.2762.

[67] C. F. Berger *et al.*, "Precise Predictions for W + 3 Jet Production at Hadron Colliders," *Phys. Rev. Lett.* **102** (2009) 222001, 0902.2760.

[68] C. F. Berger *et al.*, "Next-to-Leading Order QCD Predictions for W+3-Jet Distributions at Hadron Colliders," 0907.1984.

[69] C. F. Berger *et al.*, "NLO QCD Predictions for W+3 jets," 0909.4949.

[70] S. Dittmaier, S. Kallweit, and P. Uwer, "NLO QCD corrections to WW+jet production at hadron colliders," *Phys. Rev. Lett.* **100** (2008) 062003, 0710.1577.

[71] A. Lazopoulos, T. McElmurry, K. Melnikov, and F. Petriello, "Next-to-leading order QCD corrections to $t\bar{t}Z$ production at the LHC," *Phys. Lett.* **B666** (2008) 62–65, 0804.2220.

[72] A. Lazopoulos, K. Melnikov, and F. J. Petriello, "NLO QCD corrections to the production of $t\bar{t}Z$ in gluon fusion," *Phys. Rev.* **D77** (2008) 034021, 0709.4044.

[73] T. Binoth, G. Ossola, C. G. Papadopoulos, and R. Pittau, "NLO QCD corrections to tri-boson production," *JHEP* **06** (2008) 082, 0804.0350.

[74] R. Frederix, T. Gehrmann, and N. Greiner, "Automation of the Dipole Subtraction Method in MadGraph/MadEvent," *JHEP* **09** (2008) 122, 0808.2128.

[75] H. Baer, J. Ohnemus, and J. F. Owens, "A Next-To-Leading Logarithm Calculation of Jet Photoproduction," *Phys. Rev.* **D40** (1989) 2844.

[76] G. Kramer and B. Lampe, "Jet Cross-Sections in e+ e- Annihilation," *Fortschr. Phys.* **37** (1989) 161.

[77] K. Fabricius, I. Schmitt, G. Kramer, and G. Schierholz, "Higher Order Perturbative QCD Calculation of Jet Cross- Sections in e+ e- Annihilation," *Zeit. Phys.* **C11** (1981) 315.

[78] R. K. Ellis, D. A. Ross, and A. E. Terrano, "The Perturbative Calculation of Jet Structure in e+ e- Annihilation," *Nucl. Phys.* **B178** (1981) 421.

[79] A. Gehrmann-De Ridder and M. Ritzmann, "NLO Antenna Subtraction with Massive Fermions," 0904.3297.

[80] A. Gehrmann-De Ridder, T. Gehrmann, and E. W. N. Glover, "Quark-Gluon Antenna Functions from Neutralino Decay," *Phys. Lett.* **B612** (2005) 36–48, hep-ph/0501291.

[81] A. Gehrmann-De Ridder, T. Gehrmann, and E. W. N. Glover, "Gluon-Gluon Antenna Functions from Higgs Boson Decay," *Phys. Lett.* **B612** (2005) 49–60, hep-ph/0502110.

[82] A. Bassetto, M. Ciafaloni, and G. Marchesini, "Jet Structure and Infrared Sensitive Quantities in Perturbative QCD," *Phys. Rept.* **100** (1983) 201–272.

[83] G. Altarelli and G. Parisi, "Asymptotic Freedom in Parton Language," *Nucl. Phys.* **B126** (1977) 298.

[84] S. Dittmaier, "A general approach to photon radiation off fermions," *Nucl. Phys.* **B565** (2000) 69–122, `hep-ph/9904440`.

[85] H. Murayama, I. Watanabe, and K. Hagiwara, "HELAS: HELicity amplitude subroutines for Feynman diagram evaluations,". KEK-91-11.

[86] G. P. Lepage, "A New Algorithm for Adaptive Multidimensional Integration," *J. Comput. Phys.* **27** (1978) 192.

[87] J. Alwall *et al.*, "A standard format for Les Houches event files," *Comput. Phys. Commun.* **176** (2007) 300–304, `hep-ph/0609017`.

[88] T. Gleisberg and F. Krauss, "Automating dipole subtraction for QCD NLO calculations," *Eur. Phys. J.* **C53** (2008) 501–523, `0709.2881`.

[89] M. H. Seymour and C. Tevlin, "TeVJet: A general framework for the calculation of jet observables in NLO QCD," `0803.2231`.

[90] K. Hasegawa, S. Moch, and P. Uwer, "Automating dipole subtraction," `0807.3701`.

[91] M. Czakon, C. G. Papadopoulos, and M. Worek, "Polarizing the Dipoles," `0905.0883`.

[92] Z. Nagy and Z. Trocsanyi, "Next-to-leading order calculation of four-jet observables in electron positron annihilation," *Phys. Rev.* **D59** (1999) 014020, `hep-ph/9806317`.

[93] J. Campbell, R. K. Ellis, and F. Tramontano, "Single top production and decay at next-to-leading order," *Phys. Rev.* **D70** (2004) 094012, `hep-ph/0408158`.

[94] J. Campbell and F. Tramontano, "Next-to-leading order corrections to W t production and decay," *Nucl. Phys.* **B726** (2005) 109–130, `hep-ph/0506289`.

[95] R. Kleiss, W. J. Stirling, and S. D. Ellis, "A new Monte Carlo treatment of multiparticle phase space at high-energies," *Comput. Phys. Commun.* **40** (1986) 359.

[96] J. M. Campbell, R. Frederix, F. Maltoni, and F. Tramontano, "Next-to-Leading-Order Predictions for t-Channel Single-Top Production at Hadron Colliders," *Phys. Rev. Lett.* **102** (2009) 182003, `0903.0005`.

[97] S. Catani, "The singular behaviour of QCD amplitudes at two-loop order," *Phys. Lett.* **B427** (1998) 161–171, `hep-ph/9802439`.

[98] S. Catani, S. Dittmaier, and Z. Trocsanyi, "One-loop singular behaviour of QCD and SUSY QCD amplitudes with massive partons," *Phys. Lett.* **B500** (2001) 149–160, hep-ph/0011222.

[99] C. G. Bollini and J. J. Giambiagi, "Dimensional Renormalization: The Number of Dimensions as a Regularizing Parameter," *Nuovo Cim.* **B12** (1972) 20–25.

[100] J. F. Ashmore, "A Method of Gauge Invariant Regularization," *Lett. Nuovo Cim.* **4** (1972) 289–290.

[101] G. M. Cicuta and E. Montaldi, "Analytic renormalization via continuous space dimension," *Nuovo Cim. Lett.* **4** (1972) 329–332.

[102] W. Siegel, "Supersymmetric Dimensional Regularization via Dimensional Reduction," *Phys. Lett.* **B84** (1979) 193.

[103] W. Siegel, "Inconsistency of Supersymmetric Dimensional Regularization," *Phys. Lett.* **B94** (1980) 37.

[104] D. Stockinger, "Regularization by dimensional reduction: Consistency, quantum action principle, and supersymmetry," *JHEP* **03** (2005) 076, hep-ph/0503129.

[105] A. Signer and D. Stockinger, "Using Dimensional Reduction for Hadronic Collisions," *Nucl. Phys.* **B808** (2009) 88–120, 0807.4424.

[106] Z. Kunszt, A. Signer, and Z. Trocsanyi, "One loop helicity amplitudes for all $2 \to 2$ processes in QCD and N=1 supersymmetric Yang-Mills theory," *Nucl. Phys.* **B411** (1994) 397–442, hep-ph/9305239.

[107] J. Dai, J. F. Gunion, and R. Vega, "LHC detection of neutral MSSM Higgs bosons via g g $\to$ b anti-b h $\to$ b anti-b b anti-b," *Phys. Lett.* **B345** (1995) 29–35, hep-ph/9403362.

[108] J. Dai, J. F. Gunion, and R. Vega, "Detection of neutral MSSM Higgs bosons in four-b final states at the Tevatron and the LHC: An update," *Phys. Lett.* **B387** (1996) 801–803, hep-ph/9607379.

[109] E. Richter-Was and D. Froidevaux, "MSSM Higgs searches in multi-b-jet final states at the LHC," *Z. Phys.* **C76** (1997) 665–676, hep-ph/9708455.

[110] S. D. Ellis and D. E. Soper, "Successive combination jet algorithm for hadron collisions," *Phys. Rev.* **D48** (1993) 3160–3166, hep-ph/9305266.

[111] T. Binoth, J. P. Guillet, G. Heinrich, E. Pilon, and T. Reiter, "Golem95: a numerical program to calculate one-loop tensor integrals with up to six external legs," *Comput. Phys. Commun.* **180** (2009) 2317–2330, 0810.0992.

[112] T. Reiter, "Optimising Code Generation with haggies," 0907.3714.

[113] T. Reiter, "An Automated Approach for $q\bar{q} \to b\bar{b}b\bar{b}$ at Next-to-Leading Order QCD," 0903.4648.

[114] T. Reiter, "Automated Evaluation of One-Loop Six-Point Processes for the LHC," 0903.0947.

[115] A. Guffanti. private communication.

[116] **NLO Multileg Working Group** Collaboration, Z. Bern *et al.*, "The NLO multileg working group: Summary report," 0803.0494.

[117] A. Bredenstein, A. Denner, S. Dittmaier, and S. Pozzorini, "NLO QCD corrections to top anti-top bottom anti-bottom production at the LHC: 1. quark-antiquark annihilation," *JHEP* **08** (2008) 108, 0807.1248.

[118] G. Bevilacqua, M. Czakon, C. G. Papadopoulos, R. Pittau, and M. Worek, "Assault on the NLO Wishlist: pp $\to$ tt bb," *JHEP* **09** (2009) 109, 0907.4723.

[119] F. Campanario, C. Englert, M. Spannowsky, and D. Zeppenfeld, "NLO-QCD corrections to $W\gamma j$ production," 0908.1638.

[120] S. Dittmaier, A. Kabelschacht, and T. Kasprzik, "Polarized QED splittings of massive fermions and dipole subtraction for non-collinear-safe observables," *Nucl. Phys.* **B800** (2008) 146–189, 0802.1405.

[121] N. Greiner, "Constraints On Unparticle Physics In Electroweak Gauge Boson Scattering," *Phys. Lett.* **B653** (2007) 75–80, 0705.3518.

[122] T. Gehrmann, N. Greiner, and P. Schwaller, "Muon Excess in Cosmic Rays and at CDF: Signs for a Hidden Sector?," 0812.4240.